INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.

2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.

3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format.*

4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format.*

*For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.

8601180

Lucas, Bruce David

GENERALIZED IMAGE MATCHING BY THE METHOD OF DIFFERENCES

Carnegie-Mellon University   Ph.D.   1985

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

Copyright 1985

by

Lucas, Bruce David

All Rights Reserved

PLEASE NOTE: ·

In all cases this material has been filmed in the best possible way from the available copy.
Problems encountered with this document have been identified here with a check mark __√__ .

1. Glossy photographs or pages _____

2. Colored illustrations, paper or print _____

3. Photographs with dark background __√__

4. Illustrations are poor copy _____

5. Pages with black marks, not original copy _____

6. Print shows through as there is text on both sides of page _____

7. Indistinct, broken or small print on several pages __√__

8. Print exceeds margin requirements _____

9. Tightly bound copy with print lost in spine _____

10. Computer printout pages with indistinct print _____

11. Page(s) _____ lacking when material received, and not available from school or author.

12. Page(s) _____ seem to be missing in numbering only as text follows.

13. Two pages numbered _____. Text follows.

14. Curling and wrinkled pages _____

15. Dissertation contains pages with print at a slant, filmed as received _____

16. Other_____

_____

_____

University
Microfilms
International

Generalized Image Matching

by the

Method of Differences

Bruce D. Lucas

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

July 1984

Carnegie-Mellon University

Mellon College of Science

THESIS

submitted in partial fulfillment of the requirements

for the degree of ___Doctor of Philosophy___

Title___ GENERALIZED IMAGE MATCHING BY THE METHOD

OF DIFFERENCES

Presented By ___Bruce D. Lucas___

Accepted by the Department of___ Computer Science

Major Professor   7-21-84   Date

Department Head   7-27-84   Date

Approved By

Dean   9/16/85   Date

Abstract

Image matching refers to aligning two similar images related by a transformation such as a translation, rotation, etc. In its general form image matching is a problem of estimating the parameters that determine that transformation. These parameters may be a few global parameters or a field of parameters describing local transformations.

This thesis explores in theory and by experiment image matching by the *method of differences*. The method uses intensity differences between the images together with the spatial intensity gradient to obtain from each image point a linear constraint on the match parameters; combining constraints from many points yields a parameter estimate. The method is particularly suitable where an initial estimate of the match parameters is available. In such cases it eliminates search which can be costly, particularly in multi-dimensional parameter spaces. Essential to the technique are smoothing, which increases the range of validity of the constraint provided by the gradient, and iteration, because the parameter estimate is an approximation. Smoothing increases the range of convergence but it decreases accuracy, so a coarse-fine approach is needed. A theoretical analysis supports these claims and provides a means for predicting the algorithm's behavior.

The first application considered here, optical navigation, requires matching two images to determine the relative camera positions. Here the match parameters are the position parameters, because they determine the image transformation. In many cases, such as robot guidance, the required parameter estimate is available. Using information from points near edges minimizes error due to noise, specularity, etc. The relationship between the three-space geometry of the reference points and the stability of the algorithm is investigated. Optical navigation experiments using both real and synthetic images are presented. They support the claims of the theoretical analysis, and demonstrate a range of convergence and accuracy adequate for many tasks.

The second application, stereo vision, is a problem of determining a field of local parameters, namely the distance values. Constraints from the neighborhood of each point contribute to its distance estimate. Experiments on both real and synthetic data provide encouraging results.

Contents

Acknowledgements

This thesis owes its completion to a number of people. Foremost among these were the members of my thesis committee. Takeo Kanade, my advisor, provided much valuable guidance and showed great patience in allowing me to pursue this research. His comments on an early draft of the thesis were invaluable. Hans Moravec provided me with continual inspiration and encouragement. H. T. Kung showed admirable diligence in reading the thesis and provided many good comments, as did Dana Ballard.

Discussions with various colleagues during the course of the research were of great help. At the risk of leaving someone out, I mention a few: Larry Matthies, Chuck Thorpe, and Peter Highnam were perhaps of greatest assistance.

Finally, I would never have been able to survive the N years this endeavor took without the moral support of my good friends (I won't list them because they know who they are), and of my family.

Chapter 1

Introduction

1.1. Introduction

Image matching (or registration) is, in general terms, aligning two identical or similar images or parts of images. The subject of this thesis is a class of techniques for doing image registration by the *method of differences*. The method of differences is a matching technique that uses the image intensity gradient together with the intensity differences between the images in a procedure that iteratively improves an initial estimate. This thesis will develop a number of algorithms for various kinds of matching based on this method, and show how they can be applied to optical navigation and stereo image interpretation, both in theory and by experiment. This chapter sets the stage by discussing the importance of image registration, formally defining it and generalizing that definition, and describing the method of differences. The method of differences is compared with other methods for image matching. Then related work in this area is surveyed, and the remainder of the thesis is outlined.

1.2. Motivation

Image matching is basic to a number of vision problems. These include optical navigation (also known as motion analysis), stereo image interpretation, object analysis, change detection, and others. The first two, optical navigation and stereo image interpretation, are two of the most important; they are the two applications considered in this thesis.

Optical navigation refers to the guidance of a robot, such as a robot arm or an autonomous roving vehicle, by means of input from an optical sensor such as a camera. Robots need navigational feedback from their environment because the environment is not perfectly predictable, and because the robot's response to a command to move in a certain way is not perfectly predictable. Optical navigation is one of many ways of providing that feedback. The optical domain is particularly rich in information, but therefore correspondingly difficult for computer analysis. It is the aim of the method of differences to provide a relatively inexpensive way to do optical navigation.

The primary problem of optical navigation is this: given two views of the same scene from two different cameras, determine the parameters describing the relationship between the coordinate systems of the cameras. This problem has two essential characteristics: it is a parameter estimation problem, and in most applications a reasonable estimate of those parameters is available at the outset. As we will see, these two characteristics make this problem particularly suited to solution by the method of differences.

The objective of stereo vision (or more precisely, binocular vision) is to obtain information about the three-dimensional form of an object or a scene from two camera views. Binocular vision is one of many sources of information available about the three-dimensional form of the world. The need for reconstructing this information from camera views arises essentially due to a shortcoming in the sensors used: cameras record only a two-dimensional projection of a three-dimensional world. Thus one method of attacking this problem has been to overcome this deficiency, for example through structured lighting, sonar range detectors, contact sensors, and so on. However, all such attempts so far have not had as general applicability as vision. In addition to the practical interest in stereo, the desire to understand the human visual system has led some researchers to propose computational models of the human stereo vision process. Unfortunately, the stereo correspondence problem has proven to be extremely difficult—certainly not as easy as the facility of humans in this problem might suggest. This thesis explores another line of attack, namely the method of differences.

The difficult part of stereo, as for navigation, is the matching problem. In the case of navigation, we desire to compute the camera motion given a set of point distances. The stereo problem is the complement of this in that we wish to compute a set of point distances given the camera motion. In general, the number of points is much larger than the number of parameters of the camera motion. This formulation of the problem shows (roughly speaking) why stereo is harder than navigation: we are given less input and asked to compute more output. That is, the amount of constraint on each quantity to be computed is less.

In addition to navigation and stereo, matching has a number of other applications. For example, one method of approaching object detection in a scene might be to hypothesize the existence of an object in the scene and then to check that hypothesis by attempting to match its known appearance against the picture. In another application, Reddy & Rubin (1978) report the need to align as nearly as possible adjacent slices in lobster nerve tissue, for the purpose of mapping the neuronal connections. Finally, some researchers have looked at the possibility of using object motion detection as a means of compressing the bandwidth required for the transmission of motion picture sequences (See, for example, Limb & Murphy, 1975a, b).

1.3. Definitions

This section gives a precise definition of the image matching problem, and then generalizes that definition. Then the method of differences is described.

Preliminary definitions.   A few preliminary definitions are in order. The notation used in this thesis is discussed in Appendix A.

An image is a function $I(p)$ of a vector, p. The vector p denotes a position in the image and $I(p)$ represents the pixel value at that position. For usual images, $I(p)$ is a scalar-valued function; but we consider some "images" in which $I(p)$ is a vector-valued function. I is generally only defined over a bounded rectangular region. Often there will be two images, I_1 and I_2, under discussion.

There will frequently be a need to compare two images, so a metric of image difference will be needed. The symbol E (for error) will denote the difference between two images. The most common such metric, and the one employed in this thesis, is the L_2 norm, defined by

$$E = \sum_p (I_2(p) - I_1(p))^2. \tag{1-1}$$

Here p ranges over image points in the regions being compared. Other metrics are used in practice, but they are typically generalizations of the L_2 norm and will be mentioned as they are encountered.

Matching.   The traditional image registration problem can be defined as follows: given two images $I_1(p)$ and $I_2(p)$ related by $I_1(p) = I_2(p + h)$, determine the *disparity* vector h between them. In many real situations, the stated relationship will not hold exactly, so a slightly different formulation of the matching problem is needed: find a disparity vector h such that, as nearly as possible, $I_1(p)$ and $I_2(p + h)$ match. The degree of match is measured by some norm, such as the L_2 norm mentioned above. More formally, we want to find an h to minimize some measure of the difference between $I_1(p)$ and $I_2(p + h)$. This form of registration can be viewed as determining two global parameters, namely the components h_x and h_y of the disparity vector. Thus image matching is seen to be a parameter estimation problem.

This definition of the matching problem can be generalized in two ways: first, one can model the transformation between the images with more parameters, and solve for those parameters. For example, we could model the change between the images as a linear deformation of coordinates, that is

$$I_1(p) = I_2(pA + h),$$

where A is the matrix describing the linear deformation. (As discussed in Appendix A, we use row vectors, thus we write pA). In this case the parameters being solved for are the translation components h_x and h_y, plus the parameters characterizing the matrix A: these could be simply the entries of A if it is an unconstrained matrix, or perhaps the rotation angle if A is restricted to be a rotation matrix. In this case one is still solving for a few *global* transformation parameters. As a second generalization, we can solve for a field of *local* image transformations. In this case the image transformation is

$$I_1(p) = I_2(p + h(p)).$$

This equation allows for a disparity $h(p)$ that is different at every point p of the image. With the right choice of parameters, the former generalization allows us to do optical navigation, which is the subject of Chapters 3 and 4. The latter generalization allows us to compute a "map" of local disparities from a stereo pair, which gives us information about the three-dimensional form of the scene; this is discussed in Chapters 5 and 6.

The method of differences. We are now in a position to understand the method of differences. The method is based on the assumption that the difference between image intensities $I_1(p)$ and $I_2(p)$ at a point p can be explained, to a linear approximation, by the disparity h between the images and by the image spatial intensity gradient. The relationship is given by

$$I_1(p) - I_2(p) \approx h_x D_x I_2(p) + h_y D_y I_2(p). \tag{1-2}$$

Here, h_x and h_y are the components of the disparity vector h, and D_x and D_y denote partial differentiation with respect to x and y. This approximation will be discussed further in the next chapter. The method takes its name from the fact that it uses the image intensity differences together with the intensity gradient (which is approximated by differences) to obtain a linear constraint on the parameters being solving for, h_x and h_y.

As equation (1-2) shows, each point p results in one linear constraint. Since in this case we are solving for the two quantities h_x and h_y, we will need to combine evidence from at least two points p to obtain a unique solution. In the generalized case the chain rule will give similar linear constraints on the parameters being solved for, for example the camera motion parameters; we will need as many points as there are parameters. In practice, using a least-squares technique allows combining evidence from many more points than there are parameters, thus reducing the effects of noise and somewhat ameliorating the approximate nature of equation (1-2). This is done by minimizing the total squared deviation from the linear constraint of equation (1-2), given by

$$\sum_{P} \left(I_1(p) - I_2(p) - h_x D_x I_2(p) - h_y D_y I_2(p) \right)^2.$$

Minimization of this quantity is straightforward. The set of points p that the sum above ranges over is chosen in one of two ways: if we are solving for global parameters (such as the motion parameters), we combine the information from a set of feature points selected from the whole image. These feature points should be selected to be near edges or other similar features whose position is little affected by photometric error, such as that caused by noise and specular effects. If we are solving for a field of local parameters, we combine the evidence from each point in the neighborhood of a given point to obtain the parameters at that point.

The use of the approximation of (1-2) has two implications. First, it means that the computed parameters will only be approximate. This problem is solved by iteration: taking the computed parameters as initial guesses, use the method of differences to compare the

images as deformed by those parameters to obtain better guesses. This results in a iterative scheme, which under the right circumstances will converge to the correct values of the parameters. One concern of this thesis will be the conditions under which convergence is achieved. The second problem is that the linear approximation is a good one only over a certain range. The effect is to limit the range over which the iteration just described will converge. This problem we solve by smoothing the images; as will be shown, both in theory and by experiment, smoothing increases the range of convergence, but at the expense of accuracy. Thus we adopt a coarse-fine approach, in which very smooth images are used in the first iteration, and less smooth images in later iterations. The techniques of smoothing and iteration are fundamental to the method of differences.

1.4. Other methods

This section compares the method of differences with other parameter estimation techniques that can be used for image matching. The intent is twofold: first to show the advantages of the method of differences over other techniques, and second to put it in perspective relative to other numerical methods.

In a sense, any sort of technique that doesn't compute an exact answer in one step "searches" for a solution. However, the term *search* will be reserved here for techniques that evaluate the error function at many values of the parameter vector without using the results to determine at which parameter value to look next.

In its simplest form, search techniques compute the error function at each of a pre-specified set of parameter vectors, and choose that which minimizes the error. For example, for the error function

$$E = \sum_x \left(I_2(x + h) - I_1(x) \right)^2, \tag{1-3}$$

where the parameter is the disparity h, a search technique would merely evaluate E at for all relevant values of h and choose the h that minimizes E. The obvious drawback of such a method is that it involves many repetitions of an expensive operation, namely comparing two images. This problem is particularly exacerbated when the dimensionality of the parameter space is large, because of the large number of error-function evaluations.

More intelligent search strategies can reduce the number of points in the parameter space examined. For example, in a coarse-fine technique (Moravec, 1980), the images are matched at low resolution in a coarse search of the parameter space. On the assumption that the actual best match will be near the best match perceived at the lower resolution, a search for the best match is performed at a higher resolution in the neighborhood of the best match at the lower resolution. Even with such techniques, the number of matches that have to be tried in high-dimensional spaces is far more than practical.

By contrast, the method of differences does more work for each point in the parameter space examined, but looks at far fewer points. The method of differences can be looked at

as one of a class of techniques based on numerical analysis. All such techniques are based on minimizing the error E by solving

$$F = D_h E = 0.$$

The method of differences is a linearization technique, in that it replaces F by an approximation linear in h. This it accomplishes by replacing the exact constraint $I_2(x+h) - I_2(x) = 0$ implicit in equation (1-3) with the approximate linear constraint mentioned above, namely $I_2(x) + hD_x I_2(x) - I_1(x) \approx 0$. This makes E quadratic in h, which makes F linear in h. Further details are given in the next chapter.

Other numerical techniques are possible. For example, Newton's method approximates F by a truncated Taylor series in h. However, F has already been obtained by differentiating E w.r.t. h, so differentiating again w.r.t. h to obtain the Taylor series gives a method that requires calculating the second derivative of I_2, and so is more difficult to implement and more expensive at each iteration. Newton's method is also a linearization method. Many other numerical methods are available; Rheinboldt (1974) gives a good survey of such methods. This author is unaware of any attempts to use such methods for image matching.

The disadvantage of all iterative numerical techniques relative to search is that the numerical techniques require an initial estimate of the parameter values. In many applications, this is not a problem. For example, in robot navigation a rough idea of the robot's position can be obtained by "dead reckoning" based on the commands given to the actuators. However, even where a rough estimate is available, and thus search must be used, the method of differences provides an advantage: the search can be much coarser than it would be otherwise. The method of differences would be used at each point searched; if best match is in the neighborhood of that point, the method will converge to it, and will diverge otherwise.

1.5. Related Work

Considerable work has been done on the problem of matching. This section will concentrate only on the literature of matching techniques related to the method of differences. Thus, the work discussed here is that related to navigation and to stereo matching, and that involving difference techniques.

General matching.   Matching can been accomplished by a number of methods. These techniques can be divided into two broad classes: pixel-based techniques and feature-based techniques. In pixel-based methods, the individual pixels are in the two images are compared to determine the quality of the match. Feature-based methods compare higher-level features extracted from the images, such as edges. The method of differences is a pixel-based technique. However, as noted above, we share with edge-based techniques the desire to use points near edges for the same reason one would use edges: they provide unambiguous indications of position in the pregrep pre intro.texsence of photometric differences between the images.

The most straightforward techniques for matching involve simply computing the correlation between the images at each possible disparity and picking the best. This is the simple search technique described in the previous section. Mathematically the correlation is closely related to the L_2 norm defined in (1-1), and can be regarded as one of a number of alternatives to it. Other alternatives include the normalized correlation and the pseudo-normalized correlation of Moravec (1980). Helava (1978) provides a good discussion of the issues involved in the detection of a peak in the correlation function. He also describes the application of such matching techniques to photogrammetry.

The obvious drawback to this technique is that it is extremely computationally expensive. Various attempts have been made to improve the performance. Barnea & Silverman (1972) introduce their "sequential similarity detection algorithm" which is a correlation search method optimized by discontinuing the comparison between two image positions when it becomes apparent that they will not provide a better match than the best match found so far. This is akin in flavor to alpha-beta pruning. Moravec (1980) uses a coarse-fine strategy: the images are filtered and resampled at lower resolution; a matching is done at the lower resolution, and this coarse match is used to define a relatively small area within the next larger (higher resolution) image over which to conduct a search at that resolution. Aggarwal et al. (1981) describe a number of heuristics that can be used to speed up the search in the context of the motion understanding problem.

These techniques all assume we are interested in a relatively small set of point matches. In some cases, however, we may be interested in a dense set of matches, perhaps a match at each pixel. If our matches are represented as two-dimensional vectors describing how a particular parcel of the image has moved from one image to the other, then we have computed the optical flow field. Lee (1980) makes a good case for the importance of the optical flow field. The optical flow field represents a combination of effects due to viewer motion, object motion, object geometry, and object photometry (which describes the image intensity level seen by the camera for each part of the object under different viewing conditions). One problem is to compute the flow field; another is to extract the desired information from it.

Horn & Schunck (1981) present a technique for solving for the optical flow. At each point in the image, the image spatial intensity gradient together with the difference in intensity between the images at that point provides one linear constraint on two unknowns, that is the x and y components of the optical flow at that point. An additional constraint is needed to solve for the flow field. This part of their technique is similar to the method of differences. However, they supply the additional constraint by making a fairly weak assumption about the nature of the objects being viewed: they assume that the flow field is smooth at most points due to the coherence of matter. The smoothness constraint is formulated in terms of minimizing the difference between the flow field value at a point and the local average around that point; this introduces the extra linear constraint needed, although it also couples the flow values for nearby points; iterative techniques are used to solve the resulting large system of equations. Yachida (1983) presents a technique based

on that of Horn and Schunck with the refinement of using point matches to speed up the iteration. Cornelius & Kanade (1983) show how optical flow can be adapted for analyzing moving x-ray pictures.

Although the method of differences is unlike either of the techniques described above, both in intent and in method, it shares some features in common with them. Like the correlation methods, it uses the L_2 norm as a measure of image similarity. Also like those methods, it performs a search in a parameter space: in simple matching, the parameters are the x and y positions of the matches; in our applications, the parameters are the motion parameters of the cameras (for navigation) or the distance of a given pixel from the camera (for stereo). Unlike those methods, it uses a gradient-based iteration to direct the search. On the other hand, the method of differences shares with the optical flow algorithm of Horn and Schunck the use of the image intensity gradient together with the image intensity difference to estimate the error in the match at that point. Unlike their technique, the method of differences calculates the parameters of interest (the motion parameters or the distances) directly rather than calculating the optical flow field as an intermediate step. This allows the computational effort to be concentrated on the quantities of interest. Moreover, the Horn and Schunck technique assumes that the motion is sufficiently small that the linear approximation is accurate enough. Thus they use no smoothing, and the iteration in their method serves a different role: it is merely a technique for solving a large system of equations, which could in principle be solved directly. By way of contrast, iteration is fundamental to the method of differences and it could not in principle work without it.

Finally, Webb & Aggarwal (1983) show how to use the method of differences for doing affine matching (matching on pieces of images related by an affine transform), based on some earlier work of Lucas & Kanade (1981). Coupled with search and a shape-from-shading method, this allows them to do matching and determine object shape simultaneously.

Stereo. A number of matching methods have been proposed specifically aimed at the stereo correspondence problem. The importance of the stereo matching problem is shown by the large body of work in this field. This work comes from researchers in several fields with different interests: neurophysiologists are interested in models of stereo matching that shed light on how the process might work in the human visual system. Photogrammetrists wish to duplicate the work of human stereo photointerpreters on for example aerial photography. Robotics researchers want to develop fast stereo systems that can provide real time feedback for their robots. The techniques developed by these researchers roughly fall into three categories: disparity detector techniques, dynamic programming techniques, and correlation techniques. The method of differences most nearly fits in the third category.

In his pioneering work, Julesz (1962) introduced the notion of a collection of different disparity detectors for each point in the scene. The idea is that for each pixel for each possible disparity at that pixel there is a detector whose value in some way indicates whether the disparity at that pixel has the value that that detector represents. Julesz is also noted for his demonstration with random-dot stereograms that humans were capable of perceiving

depth in pictures devoid of any semantic content (Julesz, 1960); this provides an existence proof to vision system designers that stereo can be done at a low level.

Marr & Poggio (1976, 1979) formulated a theory of human stereopsis incorporating many of the observed facts, such as the presence of multiple spatial frequency channels. They used the idea of a collection of disparity detectors introduced by Julesz, and in addition introduced the concept of zero-crossings in bandpass-filtered images as important feature points. Thus their technique is feature based, unlike that of Julesz. Their theory made explicit the assumptions of unique disparity at each point and of smoothly varying disparity that were only implicit in previous work. Because of the extremely large number of disparity detectors involved (one for each pixel for each possible disparity), until extremely large integrated computer systems become available, the disparity detector approach is probably of most interest to psychophysiologists as a model of the human visual system.

A second approach uses a one-dimensional dynamic programming matching technique first developed for matching speech segments. This approach is based on the following observations: first, if the camera geometry is precisely known, the stereo problem reduces to matching along corresponding lines in the two images, called epipolar lines; if the cameras are exactly aligned, epipolar lines correspond to scan lines. Second, left-to-right ordering of objects is generally preserved in the two scenes. Under these conditions, features such as edges that intersect a given epipolar line can be matched against features in the other image that intersect the corresponding epipolar line using the dynamic programming technique. This technique has been used for separate scanlines by Henderson et al. (1979) and Baker & Binford (1981); Ohta & Kanade (1983) show how to extend the method to incorporate inter-scanline constraints.

The largest body of work involves systems using correlation matching. Moravec (1980) provides an example of a real robot system incorporating both stereo matching and optical navigation. The stereo matching is accomplished using a multi-resolution correlation approach. The best match found at a lower resolution is used to constrain the range of search for the match at a higher resolution. An interest operator is used to select points likely to be uniquely matchable, and then uses a pseudo-normalized pixel-based correlation to judge the quality of the match between the images.

Gennery (1980) describes a stereo vision system based on correlation matches. Given the matches, he is able to solve for all parameters relating the two cameras but the distance between them, using an iterative approach. Given an accurate a priori knowledge of the distance between the cameras he is then able to determine the exact three-space locations of the matched points. This system was designed to be used for path-planning and location determination in a roving vehicle, such as a Mars rover.

A number of variations on the basic correlation matching have been employed. Nevatia (1976) uses a correlation approach but with multiple closely spaced views (generated by the motion of the camera) to decrease the search time and the perspective distortion problem. Levine et al. (1973) developed a system intended for use on a Mars rover, using correlation

together with some heuristics to reduce the search. Yakimovsky & Cunningham (1978) describe a system in which the correlation is done not over rectangular windows but over sparser "masks."

Mori et al. (1973) attack a problem that plagues correlation techniques, that of perspective distortion: two objects (small regions in the image) will not in general have the same shape because they are seen from two different points of view. Their approach is to make an approximate match, and then distort one of the images to lessen its differences with the other, and then to iterate this procedure. Our stereo depth-map algorithm in essence does something very similar: it takes an iterative approach, and at each step of the iteration compares each point in one image with the predicted corresponding point in the other image. The method of updating the prediction at each point is different, however: the image intensity gradient is used to predict how changing the position of the match at each point will affect the quality of the match, and the effect is summarized over each small neighborhood around each point to determine how that point should be updated.

Another dimension along which stereo systems differ is whether they provide a dense disparity map or whether they provide only sparse image matches. In general, feature-based techniques fall into the latter class because features are sparse, and correlation techniques do so as well because correlation is expensive. If a depth map is desired from such technique, a method of interpolating the sparse matches is necessary. Some progress has been made in this direction, notably by Grimson (1981) and by Terzopoulos (1984). By contrast, the method of differences produces a depth map directly. This has the advantage of not requiring a complex theory of surface interpolation built up on arbitrary assumptions.

Navigation and motion detection. Less work has been done in the field of optical navigation. This work has two major thrusts. Much of the work has been concentrated on the problem of finding the parameters relating two camera positions from matches in the scenes viewed by the camera. These matches may take the form of sparse feature matches or a dense optical flow field. A second major line of research has been on object motion detection. While this is not exactly the same thing as navigation it is related in the following way: moving through a fixed environment is the same as remaining fixed and having the environment move past the observer. If we now generalize that to allow different pieces of the environment to be moving differently, we have the object motion-detection problem.

Several results concerning computing the motion parameters from a number of matches are available. Tsai & Huang (1981) present results concerning the number of point matches necessary to uniquely determine the motion parameters between two views. Gennery (1980) gives a least-squares technique that solves for all of the motion parameters (except distance between cameras) given a set of point matches using an iterative technique; uniqueness is not guaranteed. Bruss & Horn (1983) show how to uniquely determine the motion of a camera relative to its fixed environment given the optical flow field. Ballard & Kimball (1983) show how to use the Hough transform (Hough, 1962) to detect moving objects from the flow field.

The method of differences differs from these in that it directly extracts the motion parameters from the image, bypassing the steps of computing either the optical flow field or a sparse set of matching points. Either of these steps amounts to doing a matching with little or no constraint. By contrast, the method of differences combines matching with parameter estimation and thus is capable of using precisely the degree of constraint provided by the geometry of rigid body motion.

Huang & Tsai (1981) summarize the correlation and Fourier techniques for image matching, and describe the essence of the method of differences. They present a two-step method that first finds image matches (by some method) and then solves for the motion parameters. They present a direct method like the method of differences but without iteration. The method differs from the method of differences in the following way: in addition to the motion parameters, there is for each reference point used the unknown of its distance from the camera. But each point introduces only one constraint on the unknowns; therefore, the number of unknowns exceeds the number of constraints by the number of motion parameters. Their approach is to assume the points chosen lie on an object of some specified form, parameterizable by a fixed number of parameters. This has the obvious disadvantage of limited applicability. Moreover, the lack of iteration limits the accuracy and range of applicability. The method of differences assumes that the distances to the points on the objects are known. It will be argued that that this is a reasonable assumption in many applications, such as industrial robotics.

Cafforio & Rocca (1979) make a thorough and rigorous study of the effects of noise on estimates of global translation parameters by the method of differences. They then present an approximation, similar in flavor to that of Limb & Murphy (1975a, b; see below), for implementation purposes. Cafforio (1979) points out the importance of a good estimate of the image gradient.

A number of techniques have been proposed for detecting multiple moving objects. Tsai & Huang (1980) give a method based on differentials that assumes the image points used in the comparison between the pictures are images of coplanar points on the object. Their method is based on local estimates only thus can provide motion estimates for multiple moving objects. The method is quite sensitive to the initial estimate of the motion parameters, as is to be expected from the lack of smoothing and iteration.

Limb & Murphy (1975a, b) describe a difference-based technique for computing local motion parameters. It first segments the image into independent moving regions, and then estimates the motion of each region independently using a simple accumulator technique. The assumption is that the motion of each region is adequately described as a translation parallel to the image plane. Their method is particularly simple to implement in hardware. Both of these features are suited to image compression applications.

Fennema & Thompson (1979) detect multiple moving objects using a Hough-transform approach (Hough, 1962). Each image point is consistent with an object at that point whose motion is constrained to a line in velocity space: that line corresponding to velocities such

that the image intensity difference at that point is exactly accounted for by the motion and the image intensity gradient. An array of accumulators is created, each corresponding to a particular velocity. For each point in the image that has sufficient change, those accumulators corresponding to velocities consistent with the change in image intensity and the image spatial intensity gradient at that point are incremented. Peaks in the accumulator array correspond to velocities of objects in the image. They also introduce the idea of smoothing the image (by defocusing) to improve the performance of the algorithm.

1.6. Outline

Chapter 2 describes the basis of the method of differences: the difference in intensities between the two images at each point, together with the spatial intensity gradient of the image at that point, provide an estimate of how changing that match point will affect the error in the match. The gradient is in fact estimated using differences, thus the name *method of differences*. This chapter introduces the idea of *global* match parameters (such as the navigation parameters relating two cameras) and *local* match parameters (such as the distance of a point from the camera). In either case, changing one of the parameters changes the position of the matching point. Therefore information about how the match position changes with changes in parameters together with estimates of how the match quality changes with the match position allow us to estimate how the match quality changes with the match parameters, for example as illustrated by equation (1-2). This is the foundation of a gradient-based iteration that forms the heart of the method of differences. Chapter 2 discusses the method of differences in a general setting, and shows how it can be adapted to handle a variety of matching problems. The importance of smoothing and iteration to the functioning of the method are discussed. A theoretical analysis is given that helps support these claims. This theoretical analysis will help us determine the range and speed of convergence of the algorithm and to understand the effect of smoothing on the algorithm.

Chapter 3 focuses on issues related to one particular type of matching, namely matching for optical navigation. A discussion of the applications of optical navigation comes first. Then a camera model is defined and the parameters relating two cameras are enumerated. These parameters are both geometric (relating the geometry of the cameras) and photometric (relating the intensity values reported by the cameras). The navigation problem is to solve for (some subset of) these parameters. This is done by using the method of differences; the mathematics of this is developed in some detail, and the implementation of the operations is described. The net result is a system of linear equations, one for each of the unknowns. The numerical stability of this problem is of considerable interest, and this problem is investigated using theoretical tools; the primary such tool developed in this chapter is a measure of the stability of the equations due to the geometry of the reference points, as separate from the photometry of the scene. This measure is applicable to any match-based navigation technique.

Chapter 4 presents the results of a number of experiments to verify the theory of Chapter 3. The results concerning the numerical stability of the equations are tested. The

experiments investigate the range of convergence to be expected from the algorithm, both in the single- and in the multi-parameter cases, and investigate the accuracy of the result. The effect on the accuracy and range of a number of factors such as number of reference points, three-space distribution of reference points, degree of smoothing, and so on is determined. Both real and synthetic pictures are used. The main conclusion from these experiments is that the method is suitable for use in many navigation applications.

Chapter 5 is a theoretical discussion of the application of the method of differences to stereo vision. Here we wish to calculate a field of local parameters, namely the distances of points from the cameras. The method can accomplish this by matching small image patches. A coarse-fine approach is necessary, so that the remaining disparity at each resolution is no larger than approximately the size of the smoothing window at that resolution. The resulting algorithm can be implemented efficiently with a running time independent of the size of local matching window used.

Chapter 6 contains results from the stereo algorithm, applied both to real aerial photographs and to synthetic random-dot stereograms. Various parameters of the theory developed in Chapter 5 are tested. The method gives excellent results on the synthetic stereograms and promising but difficult-to-evaluate results on the natural images.

Finally, Chapter 7 summarizes the thesis, discusses its main contributions, and suggests avenues for further research. In addition, Appendix A discusses the notation used in the thesis, and Appendix B presents the algorithms used for smoothing and gradient estimation.

Chapter 2

General Algorithms

2.1. Introduction

This chapter presents a series of new algorithms for generalized image matching that use the method of differences. The different algorithms make different assumptions about the class of transformations that model the change in the image, and are of varying complexity and accuracy.

As mentioned in the introduction, the algorithms presented come in two general flavors: *global* and *local* algorithms. In the global case, one is interested in computing a set of global parameters that describe the transformation between two images, such as a global translation or rotation or scaling. In the local case, one is interested in calculating similar transformation parameters but on a local scale. That is, we assume that the neighborhood around each point in one image is mapped to a neighborhood around some point in the other image by a transformation such as translation, rotation, or scaling. In such cases we are calculating a field of parameters, such as a vector field of local translations, a field of local scalings, etc. In many cases, a given algorithm will have two very similar versions, one global and one local.

The performance of the algorithms is crucially dependent on two factors: first, since the algorithms only compute an approximation to the transformation parameters, an iterative scheme is necessary to compute the parameters accurately. Second, since such an iterative scheme requires an initial estimate of the values of the parameters, we are interested in obtaining the widest possible range of convergence of the algorithm. One way to improve convergence is to smooth the images. Since smoothing increases the range of convergence at the expense of accuracy, earlier iterations will use greater smoothing than later iterations. Using a theoretical analysis based on frequency-domain considerations, theoretical justification for intuitive arguments about the effects of smoothing and iteration is derived. In later chapters these theoretical predictions are verified.

Some of the algorithms involve combining evidence from many points (all points in the case of global parameter estimation, points in the neighborhood of each given point in

the case of local estimation) by means of a sum. Often the contribution of the evidence at each point can be weighted by a non-negative weighting function, $w(x)$. Possible weighting functions are discussed. This is largely a theoretical discussion, as the usefulness of weighting is an open question.

The remainder of this chapter will first develop basic 1-d registration algorithms, and then show how to generalize to 2-d registration problems. Then we will consider more complicated versions that model the transformation between the images in more complex ways than a simple translation. How smoothing and iteration improve the behavior of the algorithms will be discussed, and a theoretical analysis will justify these claims. This theoretical analysis will help determine the range and speed of convergence of the algorithm and to understand the effect of smoothing on the algorithm.

2.2. 1-d algorithms

1-d single point algorithm.   This form of the algorithm is the simplest and least effective, but it provides insight into more complex versions of the algorithm. It assumes two one-dimensional images, I_1 and I_2, related by

$$I_1(x) = I_2(x + h).$$

The goal is to calculate the disparity h. A truncated Taylor series provides a linear estimate of the behavior of the image near each point x:

$$I_1(x) = I_2(x + h) \approx I_2(x) + h I_2'(x); \tag{2-1}$$

thus

$$h \approx \frac{I_1(x) - I_2(x)}{I_2'(x)}. \tag{2-2}$$

The denominator of this equation refers to the derivative I_2' of I_2; but in fact an implementation will be given only samples of $I_2(x)$ at discrete values of x; thus the derivative will be approximated by a difference. This reveals the origin of the name *method of differences*: the difference between image values together with the image derivative, estimated by a difference, yields a disparity estimate

Note that this approximation depends on x; but this is in fact desirable, since we are interested in allowing the disparity h to vary with x as well, and it would be no good if the approximation did not also vary with x. To make this clearer, replace (2-1) with a generalized version, making h explicitly dependent upon x:

$$I_1(x) = I_2(x + h(x)) \approx I_2(x) + h(x) I_2'(x), \tag{2-3}$$

from which

$$h(x) \approx \frac{I_1(x) - I_2(x)}{I_2'(x)}. \tag{2-4}$$

Thus, let us define

$$\hat{h}(x) = \frac{I_1(x) - I_2(x)}{I_2'(x)}.$$ (2-5)

In general, $\hat{h}$ will denote an approximation to h, and we shall make the transition from approximations like (2-4) to equations like (2-5) without comment. It should be noted that our estimated $\hat{h}(x)$ is implicitly dependent on the actual disparity field $h(x)$, the dependency being concealed in the fact that $I_1(x) = I_2(x + h(x))$. This approximation can only be expected to be accurate at those points where h is small or where I_2 is nearly linear.

1-d average algorithm. The inaccuracy of equation (2-5) can be somewhat reduced by averaging it over some range of values of x. If we are computing a global disparity value for all of I_1 and I_2 (i.e. the "traditional" definition of disparity), the average can range over all values of x:

$$\hat{h} = \frac{1}{N} \sum_x \frac{I_1(x) - I_2(x)}{I_2'(x)}.$$ (2-6)

This formula computes a parameter $\hat{h}$ valid for all x, so it is a global algorithm. However, if a disparity map is to be computed, then the corresponding local algorithm is required. The local version of (2-6) is obtained by taking

$$\hat{h}(x) = \frac{1}{N} \sum_{x' \text{ near } x} \frac{I_1(x') - I_2(x')}{I_2'(x')},$$ (2-7)

where N is the number of points x' near (for some definition of "near") each point x. This of course corresponds to making the calculation in (2-6) for each subimage of I_1 and I_2 near x. In effect, (2-5) and (2-6) are special (limiting) cases of (2-7), where "near x" is taken to mean respectively "equal to x" and "any point at all."

1-d least squares algorithm. Unfortunately, the preceding disparity estimates suffer from two problems. First, there is no obvious way to generalize them to two dimensions; the two-dimensional linear approximation corresponding to (2-3) is

$$I_1(x, y) = I_2(x + \mathbf{h}_x(x, y), y + \mathbf{h}_y(x, y))$$

$$\approx I_2(x, y) + \mathbf{h}_x(x, y) D_x I_2(x, y) + \mathbf{h}_y(x, y) D_y I_2(x, y),$$ (2-8)

where $\mathbf{h}_x$ and $\mathbf{h}_y$ are the x and the y components respectively of the disparity field, and D_x and D_y are partial differentiation operators for x and y respectively. This equation provides a linear constraint on $\mathbf{h}_x(x, y)$ and $\mathbf{h}_y(x, y)$ at each point. Unfortunately, equation (2-8) has no unique solution for $\mathbf{h}_x(x, y)$ and $\mathbf{h}_y(x, y)$. That is, one constraint is insufficient to

determine the two quantities. In "optical flow" techniques this problem has been solved by Horn & Schunck (1981) and others by adding an additional constraint on h_x and h_y, typically for example that h_x and h_y satisfy some sort of smoothness criteria. Alternatively, one could choose that solution that is most conservative, e.g. minimizes say $h_x(x,y)^2 + h_y(x,y)^2$, but there seems to be no justification for doing this. Furthermore, neither of these solutions solve the second problem, namely that while it is clear that $\hat{h}(x)$ as defined above is in fact an approximation to $h(x)$ when $I_1(x) = I_2(x+h(x))$, it is not clear exactly what is being calculating when $I_1(x)$ is only approximately equal to $I_2(x+h(x))$.

Both of these problems are solved by a slightly different approach. Suppose (in the one-dimensional case) the goal is to compute at each point x a disparity estimate $\hat{h}(x)$ that minimizes some measure of the difference between $I_1(x')$ and $I_2(x'+\hat{h}(x))$ for each x' in a small neighborhood of x. That is, imagine computing a single disparity $\hat{h}(x)$ that will make the subpicture of I_1 near x as similar to the subpicture of I_2 near $x+\hat{h}(x)$ as possible. In the case where $I_1(x) = I_2(x+h(x))$, that will happen when and only when $\hat{h}(x) = h(x)$. We chose the L_2 norm to minimize; thus, we need to compute at each x that disparity estimate $\hat{h}(x)$ that minimizes the error function,

$$E_x = \sum_{x'\,\text{near}\,x} \left(I_2(x'+\hat{h}(x)) - I_1(x')\right)^2. \tag{2-9}$$

Making the linear approximation of (2-1), we obtain

$$E_x \approx \sum_{x'\,\text{near}\,x} \left(I_2(x') + \hat{h}(x)I_2'(x') - I_1(x')\right)^2. \tag{2-10}$$

This error measure is just the extent to which the (one-dimensional version of) the linear constraint given above in equation (2-8) is violated. This error measure will be the one generalized in a later section. Differentiating with respect to $\hat{h}(x)$ and setting equal to zero yields the estimate,

$$\hat{h}(x) = \frac{\displaystyle\sum_{x'\,\text{near}\,x} \left(I_1(x') - I_2(x')\right)I_2'(x')}{\displaystyle\sum_{x'\,\text{near}\,x} I_2'(x')^2}. \tag{2-11}$$

Note that (2-11) bears an interesting relationship to (2-7); in particular, turning (2-7) into a weighted average, where each term of the sum in (2-7) is multiplied by the "weight" $w(x') = I_2'(x')^2$, and the sum is divided by the term $\sum_{x'} w(x')$, yields (2-11). But this weighting is in fact desirable, because it gives more weight to points that are in high-gradient portions of the image; such points are less ambiguous in their position and less subject to apparent movement due to photometric noise or error. It is for much the same reason that edge-based techniques use only edges, and that the method of differences itself does best if information is taken only from reference points near edges.

1-d symmetric least squares. One objection to all the forms seen so far is that they are not symmetric with respect to I_1 and I_2. It is not clear whether this causes any problem, since the experiments reported here use only asymmetric versions of the algorithm. Nevertheless, removing the asymmetry is relatively straightforward; just allow the error to have two terms provided by

$$I_1(x) = I_2(x + h(x)) \approx I_2(x) + h(x)I_2'(x) \quad \text{and}$$

$$I_2(x) = I_1(x - h(x)) \approx I_1(x) - h(x)I_1'(x),$$

so that

$$E_x \approx \sum_{x' \text{ near } x} \left(I_2(x') + I_2'(x')h(x) - I_1(x')\right)^2$$

$$+ \sum_{x' \text{ near } x} \left(I_1(x') - I_1'(x')h(x) - I_2(x')\right)^2. \tag{2-12}$$

Differentiating with respect to $h(x)$, setting equal to zero, and solving yields a symmetrical estimate:

$$\hat{h}(x) = \frac{\displaystyle\sum_{x' \text{ near } x} \left(I_1(x') - I_2(x')\right)\left(I_2'(x') + I_1'(x')\right)}{\displaystyle\sum_{x' \text{ near } x} \left(I_2'(x')^2 + I_1'(x')^2\right)}. \tag{2-13}$$

The careful reader will note that (2-13) is in fact antisymmetric in I_1 and I_2, but this of course is desirable; interchanging I_1 and I_2 negates the disparity. Compare this with the analogous asymmetric estimate of (2-11). Whether this symmetric version provides any significant advantage over the asymmetric version is an open question.

1-d algorithms with weighting. The estimate provided by equation (2-5) varies in accuracy. Fortunately, it is possible to detect to some extent how accurate that estimate is. Figure 2-1 illustrates the principle: the disparity estimate of (2-5) is in general most accurate when the derivatives of I_1 and I_2 are nearly equal. This empirical observation is in fact justified on theoretical grounds as well:

$$I_1'(x) - I_2'(x) = I_2'(x + h) - I_2'(x) \approx hI_2''(x).$$

Thus, for a given disparity h, where the difference in derivatives is small, the second derivative of of I_2 is also small, and so the truncated Taylor series linear approximation of (2-1) is more likely to be accurate.

Thus, the effect on the final result of those points that provide an inaccurate estimate can be reduced or eliminated. Eliminating the effect of such points requires a threshold of some sort on the difference of derivatives, which leads to an ill-conditioned algorithm, in which a small change in the input data can make a large change in the result. To be
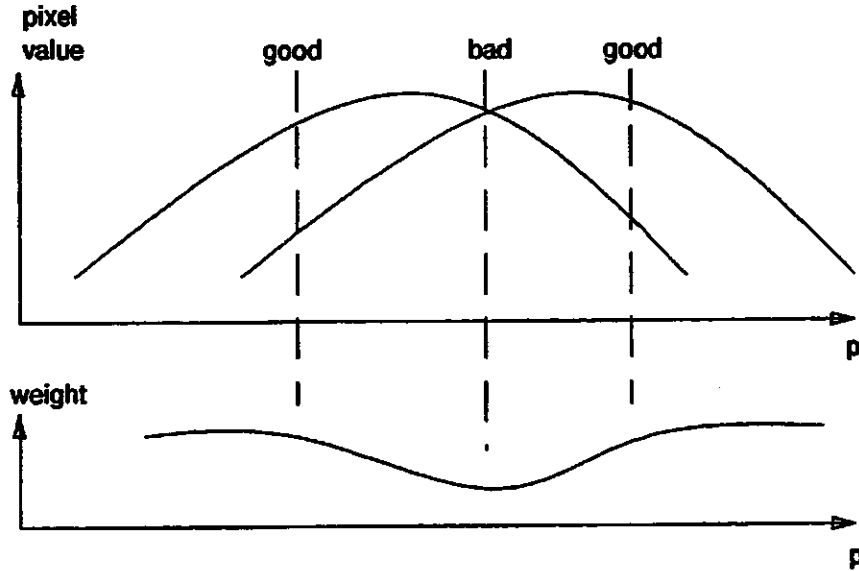
Figure 2–1. Quality of disparity estimate provided by derivatives varies from point to point as shown. Weight function takes this into account.

preferred is a scheme whereby the sums in (2–7), (2–11), or (2–13) are turned into weighted sums. For example, (2–11) becomes

$$\hat{h}(x) = \frac{\displaystyle\sum_{x' \text{ near } x} \big(I_1(x') - I_2(x')\big) I_2'(x') w(x')}{\displaystyle\sum_{x' \text{ near } x} I_2'(x')^2 w(x')}. \tag{2-14}$$

The weighting function $w(x)$ must somehow be large when the difference in derivatives is small and small when the difference is large. Several possibilities suggest themselves:

$$w(x) = 1, \tag{2-15}$$

$$w(x) = \begin{cases} 0, & \text{if } |I_2'(x) - I_1'(x)| > L, \\ 1, & \text{otherwise,} \end{cases} \tag{2-16}$$

$$w(x) = \frac{L}{L + |I_2'(x) - I_1'(x)|}, \tag{2-17}$$

$$w(x) = \frac{L}{L + \big(I_2'(x) - I_1'(x)\big)^2},$$

$$w(x) = \frac{L}{\max\left(L, |I_2'(x) - I_1'(x)|\right)},$$

$$w(x) = \frac{L}{\max\left(L, (I_2'(x) - I_1'(x))^2\right)}, \tag{2-18}$$

where L is a parameter that determines the severity of the weighting function. L also serves to normalize $w(x)$ with respect to the range of values of I_1 etc. In fact, (2-15) is equivalent to the unweighted case, and (2-16) yields the ill-conditioned threshold technique mentioned above. Weighting is thus just a general framework that subsumes both unweighted and threshold techniques.

As noted above, the sum in equation (2-11) can be considered a weighted version of the sum in (2-7). In fact, the weighting function in this case, $I_1'(x)^2$, tends to satisfy the goal that it be large in areas where the function is approximately linear. Moreover the points in regions of large gradient are more reliable in the face of photometric errors (such as noise, highlights, or response differences between cameras) because a given error in a pixel value represents a smaller error in the estimated position. Of course this does not imply that using an additional weighting function $w(x)$ in the case of (2-11) is redundant; it is possible for $I_1'(x)^2$ to be large while $w(x)$ is small and the local disparity estimate is in fact poor.

2.3. 2-d algorithms

2-d least squares. We now consider how to extend (2-11) and (2-13) to two dimensions. In the two dimensional case, a two-dimensional vector quantity, p, will take the place of the scalar x above. We will represent p as a row vector

$$\mathbf{p} = [\mathbf{p}_x \quad \mathbf{p}_y].$$

$I_1(p)$ and $I_2(p)$ will now be scalar-valued functions of a vector quantity. The disparity will be a vector-valued function of a vector, or vector a field, $h(p)$. For convenience of exposition, let us consider the two components of h separately:

$$\mathbf{h}(\mathbf{p}) = [\mathbf{h}_x(\mathbf{p}) \quad \mathbf{h}_y(\mathbf{p})].$$

Later the same result will be derived using a more compact vector notation.

The two dimensional version of the algorithm starts with a truncated two-dimensional Taylor series, already given in (2-8), and repeated here in slightly different notation:

$$I_1(\mathbf{p}) = I_2(\mathbf{p} + \mathbf{h}(\mathbf{p}))$$

$$\approx I_2(\mathbf{p}) + \mathbf{h}_x(\mathbf{p}) D_x I_2(\mathbf{p}) + \mathbf{h}_y(\mathbf{p}) D_y I_2(\mathbf{p}). \tag{2-19}$$

Here, of course, D_x means $\partial/\partial \mathbf{p}_x$ and D_y means $\partial/\partial \mathbf{p}_y$. This notation is discussed further in Appendix A. The two-dimensional version of the error equation (2-9) is

$$E_{\mathbf{p}} = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p})) - I_1(\mathbf{p}')\right)^2. \tag{2-20}$$

Here "near" means near in some two-dimensional sense. Substituting (2-19) in (2-20) yields an error estimate to (2-10), namely

$$E_{\mathbf{p}} \approx \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') + \mathbf{h}_x(\mathbf{p}) D_x I_2(\mathbf{p}') + \mathbf{h}_y(\mathbf{p}) D_y I_2(\mathbf{p}') - I_1(\mathbf{p}) \right)^2. \tag{2-21}$$

As promised above, this error measure is just the extent to which the linear constraint of equation (2-8) or (2-19) is violated. Differentiating with respect to $\mathbf{h}_x(\mathbf{p})$ and $\mathbf{h}_y(\mathbf{p})$ results in linear equations in two unknowns,

$$\mathbf{h}_x(\mathbf{p})\Sigma_1 + \mathbf{h}_y(\mathbf{p})\Sigma_2 + \Sigma_3 = 0$$

$$\mathbf{h}_x(\mathbf{p})\Sigma_4 + \mathbf{h}_y(\mathbf{p})\Sigma_5 + \Sigma_6 = 0,$$

where

$$\Sigma_1 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(D_x I_2(\mathbf{p}') \right)^2, \qquad \Sigma_3 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') - I_1(\mathbf{p}') \right) D_x I_2(\mathbf{p}'),$$

$$\Sigma_5 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(D_y I_2(\mathbf{p}') \right)^2, \qquad \Sigma_6 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') - I_1(\mathbf{p}') \right) D_y I_2(\mathbf{p}'),$$

$$\Sigma_2 = \Sigma_4 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} D_x I_2(\mathbf{p}') \, D_y I_2(\mathbf{p}').$$

The solution is of course

$$\mathbf{h}_x(\mathbf{p}) = \frac{\Sigma_5 \Sigma_3 - \Sigma_2 \Sigma_6}{\Sigma_2^2 - \Sigma_1 \Sigma_5},$$

$$\mathbf{h}_y(\mathbf{p}) = \frac{\Sigma_2 \Sigma_3 - \Sigma_1 \Sigma_6}{\Sigma_2^2 - \Sigma_1 \Sigma_5}. \tag{2-22}$$

The same result can be written in a more compact vector notation. Rewrite (2-19) as

$$I_1(\mathbf{p}) \approx I_2(\mathbf{p}) + \mathbf{h}(\mathbf{p}) D_{\mathbf{p}} I_2(\mathbf{p});$$

$D_{\mathbf{p}}$ is the gradient operator with respect to $\mathbf{p}$, that is

$$D_{\mathbf{p}} = [\frac{\partial}{\partial \mathbf{p}_x} \quad \frac{\partial}{\partial \mathbf{p}_y}]^{\mathsf{T}}$$

(see Appendix A). Rewrite equation (2-21) as

$$E_{\mathbf{p}} \approx \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}) + \mathbf{h}(\mathbf{p}) D_{\mathbf{p}} I_2(\mathbf{p}) - I_1(\mathbf{p}) \right)^2.$$

To minimize, set

$$0 = D_{\mathbf{h}} E_{\mathbf{p}} \approx \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') - I_1(\mathbf{p}') \right) D_{\mathbf{p}} I_2(\mathbf{p}')$$

$$+ \mathbf{h}(\mathbf{p}) \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(D_{\mathbf{p}} I_2(\mathbf{p}') \right) \left(D_{\mathbf{p}} I_2(\mathbf{p}') \right)^{\mathsf{T}}.$$

But the sum that is multiplied by $\mathbf{h}$ is a matrix; solving this equation is equivalent to solving the system of linear equations derived above.

2-d symmetric algorithms. A symmetric 2-d version of the algorithm is possible. We change (2–21) into a form analogous to (2–12):

$$E_{\mathbf{p}} = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') + h_x(\mathbf{p}) D_x I_2(\mathbf{p}') + h_y(\mathbf{p}) D_y I_2(\mathbf{p}') - I_1(\mathbf{p}') \right)^2$$

$$+ \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_1(\mathbf{p}') - h_x(\mathbf{p}) D_x I_1(\mathbf{p}') - h_y(\mathbf{p}) D_y I_1(\mathbf{p}') - I_2(\mathbf{p}') \right)^2.$$

Differentiating with respect to $h_x(\mathbf{p})$ and $h_y(\mathbf{p})$ and setting equal to zero results in the two equations

$$h_x(\mathbf{p})\Sigma_1 + h_y(\mathbf{p})\Sigma_2 + \Sigma_3 = 0$$

$$h_x(\mathbf{p})\Sigma_4 + h_y(\mathbf{p})\Sigma_5 + \Sigma_6 = 0,$$

where

$$\Sigma_1 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(D_x I_2(\mathbf{p}') \right)^2 + \left(D_x I_1(\mathbf{p}') \right)^2,$$

$$\Sigma_2 = \Sigma_4 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} D_x I_2(\mathbf{p}') \, D_y I_2(\mathbf{p}') + D_x I_1(\mathbf{p}') \, D_y I_1(\mathbf{p}'),$$

$$\Sigma_3 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') - I_1(\mathbf{p}') \right) \left(D_x I_2(\mathbf{p}') + D_x I_1(\mathbf{p}') \right),$$

$$\Sigma_5 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(D_y I_2(\mathbf{p}') \right)^2 + \left(D_y I_1(\mathbf{p}') \right)^2,$$

$$\Sigma_6 = \sum_{\mathbf{p}' \text{ near } \mathbf{p}} \left(I_2(\mathbf{p}') - I_1(\mathbf{p}') \right) \left(D_y I_2(\mathbf{p}') + D_y I_1(\mathbf{p}') \right).$$

The solution of course is the same as (2–22), but with different values for Σ_1, etc. Again, this result can be expressed in a more compact vector notation.

2-d weighting. As with the one-dimensional case, weighting, smoothing, and iteration improve the performance of the algorithm. The weighting functions analogous to (2–15) through (2–18), with one addition, are

$$w(\mathbf{p}) = 1,$$

$$w(\mathbf{p}) = \begin{cases} 0, & \text{if } |D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p})| + |D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p})| > L, \\ 1, & \text{otherwise,} \end{cases}$$

$$w(\mathbf{p}) = \begin{cases} 0, & \text{if } (D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p}))^2 + (D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p}))^2 > L, \\ 1, & \text{otherwise,} \end{cases}$$

$$w(\mathbf{p}) = \frac{L}{L + |D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p})| + |D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p})|},$$

$$w(\mathbf{p}) = \frac{L}{L + \left(D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p})\right)^2 + \left(D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p})\right)^2},$$

$$w(\mathbf{p}) = \frac{L}{\max\left(L, |D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p})| + |D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p})|\right)},$$

$$w(\mathbf{p}) = \frac{L}{\max\left(L, (D_x I_2(\mathbf{p}) - D_x I_1(\mathbf{p}))^2 + (D_y I_2(\mathbf{p}) - D_y I_1(\mathbf{p}))^2\right)}.$$

2.4. More general image transformations

This section shows how three more general image transformations can be accommodated by the method of differences. First we consider images in which not only a geometric but also a photometric transformation exists between the images; then we consider images that are related by an affine transformation; and finally we see how global parameters and local parameters can be solved for simultaneously. This section is not intended to be an exhaustive inventory of applications of the method, but rather is intended to demonstrate the versatility of the method of differences.

1-d with photometric parameters. It is frequently the case that one image of a pair does not have the same contrast or the same brightness as the other. This can result from differences in lighting conditions, sensors, or photographic processing. In such cases, the difference between the images could be reasonably modeled as a linear transformation of intensity levels, on top of some transformation of coordinates, $T(x)$:

$$\gamma I_1(x) + \beta \approx I_2(T(x)). \tag{2-23}$$

The parameters of the linear transformation of intensities, β and γ, can be thought of as brightness (bias) and contrast (gain), respectively.

By way of illustration, consider the one-dimensional translational case, where $T(x)$ is given by $x + h$. The appropriate error function to minimize is

$$E \approx \sum_x \left(I_2(x) + h I_2'(x) - \gamma I_1(x) - \beta\right)^2.$$

Note that this equation treats h, β, and γ as global parameters. Differentiating with respect to h, β, and γ yields

$$h\Sigma_1 + \gamma\Sigma_2 + \beta\Sigma_3 + \Sigma_4 = 0$$

$$h\Sigma_5 + \gamma\Sigma_6 + \beta\Sigma_7 + \Sigma_8 = 0 \tag{2-24}$$

$$h\Sigma_9 + \gamma\Sigma_{10} + \beta\Sigma_{11} + \Sigma_{12} = 0,$$

where

$$\Sigma_1 = \sum_x I_2'(x)^2, \qquad \Sigma_2 = \Sigma_5 = \sum_x -I_2'(x)I_1(x),$$

$$\Sigma_3 = \Sigma_9 = \sum_x -I_2'(x), \qquad \Sigma_4 = \sum_x I_2'(x)I_2(x),$$

$$\Sigma_6 = \sum_x I_1(x)^2, \qquad \Sigma_7 = \Sigma_{10} = \sum_x I_1(x),$$

$$\Sigma_8 = \sum_x -I_1(x)I_2(x), \qquad \Sigma_{11} = \sum_x 1, \qquad \Sigma_{12} = \sum_x I_2(x),$$

whose solution is straightforwardly obtained by solving the linear equations given in (2–24). Again, a more compact vector notation is possible.

Affine registration. The method of differences can also be used to estimate global motion parameters, such as rotation, scaling etc. To illustrate the technique, let us assume that I_1 is related to I_2 by an affine transformation of the coordinates, that is

$$I_1(\mathbf{p}) = I_2(\mathbf{p}A + \mathbf{h}),$$

where A is an arbitrary matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

There are two reasons for considering such transformations of image coordinates. The first is that in many situations, rigid-body motions in the scene result in transformations that in image coordinates are reasonably approximated by affine transformations or even by more restricted image-coordinate transformations such as rotation and scaling. In particular, rigid-body motions of planar point sets result in affine transformations in the image under orthography. The second reason is that it is instructive to consider this simpler case first.

The appropriate linear approximation is

$$I_2\big(\mathbf{p}(A + \Delta A) + (\mathbf{h} + \Delta \mathbf{h})\big) \approx I_2(\mathbf{p}A + \mathbf{h}) + (\mathbf{p}\Delta A + \Delta \mathbf{h})D_{\mathbf{p}}I_2(\mathbf{p}A + \mathbf{h}),$$

where $D_{\mathbf{p}}$ is the gradient operator with respect to $\mathbf{p}$, as a column vector. This is expressed in a form that allows calculating ΔA and $\Delta \mathbf{h}$, under the assumption that it will be used in an iterative scheme. (Iteration in general is discussed in the following section). If we use $\mathbf{h}$ instead of $\mathbf{h} + \Delta \mathbf{h}$, we are implicitly taking the initial guess for $\mathbf{h}$ to be 0, which is done in later iteration schemes (equation (2–32)). On the other hand, the initial guess for A must be something more like the identity matrix, so we might have $I + A$, with A initially zero. Instead, we chose to have $A + \Delta A$, with A initially the identity.

The corresponding error quantity to minimize is

$$E = \sum_{\mathbf{p}} \left(I_2(\mathbf{p}A + \mathbf{h}) + (\mathbf{p}\Delta A + \Delta \mathbf{h})D_{\mathbf{p}}I_2(\mathbf{p}A + \mathbf{h}) - I_1(\mathbf{p})\right)^2. \qquad (2\text{-}25)$$

Now, this formula is quadratic in the quantities to be minimized with respect to, namely the components of the matrix ΔA and the vector $\Delta \mathbf{h}$. Carrying out the differentiations with respect to these quantities and solving the resulting linear equations is straightforward but tedious. The result, as in previous cases is a set of bilinear forms in the second-order statistics of the image and its derivatives. Previous remarks about smoothing, iteration, and weighting apply.

Making A less general allows it to represent subsets of affine transformations such as rotations, scalings, etc. Restricting the class of A being solved for may require a further linear approximation in order to solve (2-25). For example, suppose we want to consider only rotations Then A has the form

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where we wish to solve for θ. Now (2-25) is no longer quadratic in the parameter being solved for, namely $\Delta\theta$, so we must replace ΔA in (2-25) by the approximation

$$\Delta A \approx \Delta\theta \frac{\partial A}{\partial \theta} = \Delta\theta \begin{bmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix}.$$

Thus (2-25) becomes

$$E = \sum_{\mathbf{p}} \left(I_2(\mathbf{p}A + \mathbf{h}) + \Delta\theta(\mathbf{p}\frac{\partial A}{\partial \theta} + \Delta \mathbf{h})D_{\mathbf{p}}I_2(x) - I_1(x)\right)^2.$$

which is once again quadratic in the parameters to be estimated, namely $\Delta\theta$ and the components of $\Delta \mathbf{h}$.

Mixed global and local parameter estimation. One may wish to simultaneously solve for a few global parameters and a field of local parameters. Taking a simple example, in the one-dimensional case, suppose we wish to calculate a global brightness adjustment β and a field of disparity values $h(x)$; that is, assume, like in (2-23), that

$$I_1(x) + \beta = I_2(x + h(x)),$$

so that

$$E_x = \sum_{x' \text{ near } x} \left(I_2(x' + h(x)) - I_1(x') - \beta\right)^2, \quad \text{and} \qquad (2\text{-}26)$$

$$E = \sum_{x} \left(I_2(x + h(x)) - I_1(x) - \beta \right)^2. \tag{2-27}$$

E_x represents the local error due to each $h(x)$, and E represents the global error due to β. Thus, if we simultaneously minimize (2-26) with respect to $h(x)$ and (2-27) with respect to β, then we can say that no local error E_x could be improved by adjusting $h(x)$ and at the same time the global error E cannot be improved by adjusting β. Substituting $I_2(x) + h(x)I_2'(x)$ for $I_2(x + h(x))$ and differentiating (2-26) w.r.t. $h(x)$ and (2-27) w.r.t. β and setting to zero yields

$$0 = \sum_{x' \text{ near } x} I_2'(x')(I_2(x') - I_1(x')) + h(x) \sum_{x' \text{ near } x} I_2'(x')^2 + \beta \sum_{x' \text{ near } x} I_2'(x'), \tag{2-28}$$

$$0 = \sum_{x} I_2(x) - I_1(x) + \sum_{x} h(x)I_2'(x) - \beta \sum_{x} 1. \tag{2-29}$$

Equation (2-28) really represents as many equations as there are pixels in the image, say N. Likewise, $h(x)$ represents N different variables to solve for, one for each value of x. Thus there are $N + 1$ linear equations in $N + 1$ unknowns (the $h(x)$ and β). These equations can be solved by the following strategy. For clarity, let us $\sum$ to mean $\sum_x$ and $\sum'$ to mean $\sum_{x' \text{ near } x}$, and drop the argument from $I_1(x)$, etc., letting the nearest summation indicate what the argument is. Solving (2-28) for $h(x)$ yields

$$h(x) = \frac{-\sum' I_2'(I_2 - I_1) - \beta \sum' I_2'}{\sum' I_2'^2}. \tag{2-30}$$

Note that this is linear in β, so that substituting $h(x)$ from (2-30) into (2-29) results in a linear equation in β, which we can solve, obtaining

$$\beta = -\frac{\sum I_2' \left(\sum' I_2'(I_2 - I_1) / \sum' I_2'^2 \right)}{\sum \left(1 + I_2' \sum' I_2' / \sum' I_2'^2 \right)}. \tag{2-31}$$

Finally, we can substitute this expression for β into (2-30), obtaining an expression for $h(x)$. In practice, we would obtain a numerical value for β from (2-31) and substitute this numerical value into (2-30) to compute the values for $h(x)$.

2.5. Improving the performance

As the algorithms are presented above, they have a very limited range of applicability. This is because first, the answer computed is only an estimate; and second, the linearity assumption limits the range of disparities over which the technique produces a reasonable estimate. Iteration and smoothing respectively address these two issues.

Iteration. Since the method of differences is based on an approximation, the answer it yields is only an estimate. Given that one expects the estimates to be more accurate for smaller disparities, and provided that the estimate at least has the right sign, then it should be possible to "move" I_2 relative to I_1 by the estimated disparity and compute a new disparity estimate, which is then added to the first. This yields an iterative scheme. For the local estimation case, where a disparity field $h(x)$ is being computed, the iteration is formally given by

$$\hat{h}_0(x) = 0, \qquad \forall x,$$

$$\hat{h}_k(x) = \hat{h}_{k-1}(x) + \hat{h}(x)\Big|_{I_2(x)=I_2(x+\hat{h}_{k-1}(x))}. \tag{2-32}$$

The notation in the second line means "$\hat{h}(x)$ (as defined for example by (2-5), (2-7), etc.), with $I_2(x)$ replaced by $I_2(x+\hat{h}_{k-1}(x))$." This notation is required because the dependence of $\hat{h}(x)$ on I_2 is only implicit. For example, for the definition of $\hat{h}$ given in (2-5), the iterative version is

$$\hat{h}_0(x) = 0, \qquad \forall x,$$

$$\hat{h}_k(x) = \hat{h}_{k-1}(x) + \frac{I_1(x) - I_2(x+\hat{h}_{k-1}(x))}{I_2'(x+\hat{h}_{k-1}(x))}.$$

An iteration scheme for a global disparity estimate $\hat{h}$ is obtained from (2-32) by replacing $\hat{h}(x)$, $\hat{h}_k(x)$, etc. with $\hat{h}$, $\hat{h}_k$, etc. The following section considers theoretical conditions for the convergence of this iteration, and Chapter 4 presents experimental results.

Smoothing. Because the method of differences is based on a linear approximation, one would expect that the method works best in areas of the images that are nearly locally linear, where "locally" means on a scale proportional to the actual disparity, and worst when the images are highly non-linear. Smoothing an image tends to make it locally linear, where "locally" here means on a scale proportional to the size of the smoothing window (impulse response function of the smoothing filter). This suggests that we should smooth the images first with a smoothing window whose size is proportional to the expected disparity. It will be shown both theoretically in the following sections and empirically in subsequent chapters that smoothing the images does in fact improve the performance of these algorithms.

Smoothing has two problems. First, it distorts the images so that the computed match is not accurate. This is solved in conjunction with iteration by using a coarse-fine approach: earlier iterations, where the current estimate is less accurate, use a large smoothing window; but as the estimate becomes more and more accurate, smaller and smaller smoothing windows are used. Second, there is obviously a limit on the size of the smoothing window: if it is much larger than the image, all detail from the image will be smoothed out, and no result could be obtained. But the only reason one would want to use such a window is if the

expected disparity were greater than the size of the image, or equivalently that the match parameter estimates were so bad that the predicted match positions were off the image. Thus the restriction that the smoothing window not be too large implies that a reasonable starting estimate of the parameters must be available.

The following sections present a theoretical analysis of the algorithms that will aid in the analysis of the criteria for convergence in a later section.

2.6. Performance analysis

This and the following sections analyze the accuracy of the parameter estimates provided by the method of differences and the convergence behavior of algorithms based on the method. This analysis makes some idealizing assumptions about I_1 and I_2, and furthermore only considers the simple one-parameter registration problem. Nevertheless, it provides valuable insight into the behavior of the algorithms in the multi-parameter case, and its predictions are largely vindicated by experimental evidence.

Consider the one-dimensional case, i.e. a pair of "images" $I_1(x)$ and $I_2(x)$ are given, and suppose it is known that

$$I_1(x) = I_2(x + h).$$

Then the following formula provides an estimate $\hat{h}$ of the global disparity h:

$$\hat{h} = \frac{\sum_x (I_1(x) - I_2(x)) I_2'(x)}{\sum_x I_2'(x)^2}. \tag{2-33}$$

This is the global version of (2-11). The following section will analyze the relationship between $\hat{h}$ and h, and show how it depends on the nature of the image I_2. A better understanding of the behavior of $\hat{h}$ results from an analysis of a continuous version of (2-33).

Continuous case. For the moment, let us assume that I_1 and I_2 are defined at all points, not just at discrete sample points. As will be shown below, this simplifies the analysis but does not materially alter it. Furthermore, because the continuous case is the limit of the discrete case as the number of sample points goes to infinity, it is for many situations is a useful approximation and provides better understanding of the more complicated discrete case. Also assume that the range of interest of the function is for x between 0 and 1. Outside of this range the function will be extended by periodicity. This is obviously a simplifying assumption, and it is worse when the disparity h is large. The continuous version of (2-33) is

$$\hat{h} = \frac{\int_0^1 (I_1(x) - I_2(x)) I_2'(x)\, dx}{\int_0^1 I_2'(x)^2\, dx}. \tag{2-34}$$

Our attention will now be directed toward deriving an equivalent expression for $\hat{h}$ that is more enlightening.

Now, $I_2(x)$ can be represented on the interval 0 to 1 by a Fourier series:

$$I_2(x) = \sum_{k \geq 0} r_k \cos(2\pi kx + \theta_k), \tag{2-35}$$

where the r_k's represent the (square-root of) the power spectrum, and the θ_ks represent the phase spectrum. Representing the function in this form makes it periodic outside the interval $[0, 1]$; this is one respect in which our analysis is only approximate. (Note that θ_0 is undefined, in that changing θ_0 only scales its cosine term which can be compensated for by scaling r_0; for definiteness, take θ_0 to be 0.) Equation (2-35) implies that

$$I_1(x) = I_2(x + h) = \sum_{k \geq 0} r_k \cos(2\pi kx + 2\pi kh + \theta_k), \tag{2-36}$$

$$I_2'(x) = \sum_{k \geq 0} r_k 2\pi k \cos(2\pi kx + \theta_k + \frac{\pi}{2}), \text{ and} \tag{2-37}$$

$$I_1'(x) = \sum_{k \geq 0} r_k 2\pi k \cos(2\pi kx + 2\pi kh + \theta_k + \frac{\pi}{2}). \tag{2-38}$$

Furthermore, (2-34) can be represented using inner products by

$$\hat{h} = \frac{\langle I_1(x) - I_2(x), I_2'(x) \rangle}{\langle I_2'(x), I_2'(x) \rangle}. \tag{2-39}$$

The notation $\langle f(x), g(x) \rangle$ represents the inner product of functions f and g, which is defined by

$$\langle f(x), g(x) \rangle = \int_0^1 f(x)\overline{g(x)}\, dx,$$

where the overbar denotes the complex conjugate; since all our functions will be real, this can be neglected. The inner product of functions is used in this analysis because of a number of useful properties, namely

$$\langle f(x), g(x) + h(x) \rangle = \langle f(x), g(x) \rangle + \langle f(x), h(x) \rangle \tag{2-40}$$

$$\langle f(x) + g(x), h(x) \rangle = \langle f(x), h(x) \rangle + \langle g(x), h(x) \rangle$$

$$\langle af(x), bg(x) \rangle = a\bar{b}\, \langle f(x), g(x) \rangle$$

$$\langle \cos(2\pi kx + \theta_k), \cos(2\pi lx + \phi_l) \rangle = \begin{cases} \frac{1}{2}\cos(\theta_k - \phi_l), & \text{if } k = l \\ 0, & \text{if } k \neq l \end{cases} \tag{2-41}$$

$\langle f(x), f'(x) \rangle = 0, \quad$ for f periodic

$$\langle f(x+a), g(x+a) \rangle = \langle f(x), g(x) \rangle. \tag{2-42}$$

Equation (2-41) holds only if k and l are integers; this equation is of course motivated by our representation of I_1 etc. as cosine Fourier series.

Using these rules, equation (2-39) can be rewritten (dropping the arguments for clarity):

$$\hat{h} = \frac{\langle I_1, I_2' \rangle - \langle I_2, I_2' \rangle}{\langle I_2', I_2' \rangle} = \frac{\langle I_1, I_2' \rangle}{\langle I_2', I_2' \rangle}. \tag{2-43}$$

Finally, simplifying this using the formulas given above yields

$$\hat{h} = \frac{\displaystyle\sum_{k \geq 0} r_k^2 k \sin 2\pi k h}{2\pi \displaystyle\sum_{k \geq 0} r_k^2 k^2}. \tag{2-44}$$

This is the desired expression of the estimated disparity, $\hat{h}$, in terms of the actual disparity h and the image power spectrum r_k^2. The numerator is in fact itself a Fourier series where each term has phase 0; the denominator is a normalizing factor. The phase information disappears when equation (2-41) (first case) is applied to the numerator of equation (2-43).

This is a very interesting result, because it says that the estimated disparity $\hat{h}$ depends only on the power spectrum of the image I_2 and is independent of the phase spectrum. What makes this so interesting is that the power spectra of different images tend to be very similar, characterized by one or two parameters, while all the information of a picture is contained in the phase spectrum (Kretzmer 1952, Helava 1978, Oppenheim & Lim 1981). In particular, the power spectra are dominated by the low frequency terms with something like an exponential dropoff in power with increasing frequency k. One such characterization is

$$r_k = e^{ak+b}. \tag{2-45}$$

Ideally, one would like to characterize the behavior of $\hat{h}$ in terms of the parameters a and b by substituting (2-45) into (2-44); unfortunately the mathematics seems intractable. Even if such a result were possible, it would be only of limited interest because the parameterization of the power spectrum is of course only approximate, and because of the approximate nature of the analysis itself.

Three alternatives to an analytical solution are possible. First, one could obtain typical values of a and b from images, substitute them into (2-45) and (2-44), and numerically plot the estimate $\hat{h}$ as function of the disparity h. Second, one could obtain typical power spectra from images and substitute them into (2-44) and again plot the results numerically. Third, one could apply the algorithm to actual images and plot the results. The first approach

suffers from the inaccuracy of both (2–45) and (2–44); the second suffers only from the inaccuracy of (2–44); and the third suffers from neither problem. All three approaches amount to exploring the space of results defined in ideal form by equations (2–45) and (2–44). This space is a family of curves of disparity estimates $\hat{h}$ vs. actual disparities h, approximately parameterized by the a and b of equation (2–45). Since the third approach is functionally equivalent to the first two but is more accurately related to actual images, it is the one followed here. The results of these experiments are reported in the next chapter.

Moreover, as a later section shows, from equation (2–44) one can still make some fairly general claims about the behavior of the algorithm under smoothing. This is because smoothing acts to attenuate the higher frequency terms of the Fourier series, and equation (2–43) provides a general understanding of why this helps.

First, however, the next section derives the discrete analog of equation (2–44). This is primarily for completeness, and the reader may skip to the following section for a discussion the consequences of this equation with respect to smoothing and iteration.

Discrete case. There are four objections on theoretical grounds to the above result. First, since in most practical cases we only have discrete samples of the functions, the integrals of (2–34) cannot be computed, but rather only the sums of (2–33). Second, again because we only have discrete samples of the functions, the derivatives $I_1'(x)$ and $I_2'(x)$ cannot be computed, but only discrete differences. Third, equations (2–37) and (2–38) are only valid if the periodic extension of I_2 (defined by the equation (2–35)) is continuously differentiable. This requires that $I_2(1) = I_2(0)$, a rather considerable restriction. These first three objections will be answered by resorting to a discrete form of the analysis given above, which does not substantially change the result.

The fourth objection is that the use of the Fourier series in (2–36) to represent I_1 makes the assumption that I_2 is periodic with period 1, that is for example that $I_1(1) = I_2(1 + h) = I_2(h)$. However, this assumption is no problem when $h = 0$; as h increases, the region over which I_1 is incorrectly represented increases, and is in fact of size h. Thus, for small h the analysis will be approximately correct, only failing as h becomes larger.

Let us now proceed to the discrete analysis of (2–33). The discrete Fourier representation analogous to (2–35) is

$$I_2(x) = \sum_{k=0}^{N/2} r_k \cos(2\pi k x + \theta_k), \tag{2-46}$$

where the r_k and θ_k of (2–46) are different from the r_k and θ_k of (2–35). Equation (2–46) only holds for $x = 1/N, 2/N, \ldots, (N-1)/N$ (Note that both θ_0 and $\theta_{N/2}$ are undefined, in that changing either only scales their respective cos terms at the sample points; for definiteness, take both to be 0.) Correspondingly, the discrete form of (2–36) is

$$I_1(x) = I_2(x + h) = \sum_{k=0}^{N/2} r_k \cos(2\pi k x + 2\pi k h + \theta_k).$$

In addition to the above change, (2-37) and (2-38) must be changed to reflect the change from continuous derivative to discrete difference. In keeping with the spirit of using the same symbol for discrete and continuous inner products, we also use the prime to indicate the discrete (normalized) difference. Consider two definitions of this difference, the forward and centered differences, given by

$$
f'(x) = \begin{cases} \dfrac{1}{\epsilon}\big(f(x+\epsilon) - f(x)\big) & \text{(forward)}, \\[3mm] \dfrac{1}{2\epsilon}\big(f(x+\epsilon) - f(x-\epsilon)\big) & \text{(centered)}, \end{cases}
$$

where $\epsilon = 1/N$. Again, in either case, the continuous derivative is the limit of the difference as N goes to infinity. The differences for the function $\cos(ax + \theta)$ are given by

$$
\big(\cos(ax+\theta)\big)' = \begin{cases} \dfrac{1}{\epsilon}\big(\cos(a(x+\epsilon)+\theta) - \cos(ax+\theta)\big) \\[2mm] \quad = a\,\rho(a\epsilon)\cos(ax+\theta+\alpha(a\epsilon)) & \text{(forward)}, \\[4mm] \dfrac{1}{2\epsilon}\big(\cos(a(x+\epsilon)+\theta) - \cos(a(x-\epsilon)+\theta)\big) \\[2mm] \quad = a\,\mathrm{sinc}(a\epsilon)\cos\big(ax+\theta+\dfrac{\pi}{2}\big) & \text{(centered)}, \end{cases}
$$

where

$$
\rho(t) = \left|\frac{e^{it}-1}{t}\right| = \sqrt{\frac{2 - 2\cos t}{t^2}},
$$

$$
\alpha(t) = \arg(e^{it} - 1) = \tan^{-1}\frac{\sin t}{\cos t - 1}, \text{ and}
$$

$$
\mathrm{sinc}(t) = \frac{\sin t}{t}.
$$

Thus, (2-37) and (2-38) become respectively

$$
I_2'(x) = \begin{cases} \displaystyle\sum_{k=0}^{N/2} r_k 2\pi k\,\rho\Big(\frac{2\pi k}{N}\Big)\cos\Big(2\pi kx + \theta_k + \alpha\big(\frac{2\pi k}{N}\big)\Big) & \text{(forward)}, \\[5mm] \displaystyle\sum_{k=0}^{N/2} r_k 2\pi k\,\mathrm{sinc}\Big(\frac{2\pi k}{N}\Big)\cos\Big(2\pi kx + \theta_k + \frac{\pi}{2}\Big) & \text{(centered)}, \end{cases}
$$

$$
I_1'(x) = \begin{cases} \displaystyle\sum_{k=0}^{N/2} r_k 2\pi k\,\rho\Big(\frac{2\pi k}{N}\Big)\cos\Big(2\pi kx + 2\pi kh + \theta_k + \alpha\big(\frac{2\pi k}{N}\big)\Big) & \text{(forward)}, \\[5mm] \displaystyle\sum_{k=0}^{N/2} r_k 2\pi k\,\mathrm{sinc}\Big(\frac{2\pi k}{N}\Big)\cos\Big(2\pi kx + 2\pi kh + \theta_k + \frac{\pi}{2}\Big) & \text{(centered)}. \end{cases}
$$

For its part, the discrete inner product is defined by

$$\langle f(x), g(x)\rangle_N = \frac{1}{N}\sum_x f(x)\overline{g(x)},$$

where the summation ranges over a set of N equally spaced points on the interval from 0 to 1. The subscript N on the discrete inner product symbol makes it clear that this is a different operator from the continuous inner product; but the subscript will often be omitted because the intended inner product will be clear from the context. Note that the continuous inner product is just the limit of the discrete inner product as N goes to infinity. Thus, equation (2–39) represents equation (2–33) just as well as it does equation (2–34), given the proper interpretation of the inner product symbol. Furthermore, the discrete inner product still obeys equations (2–40) through (2–42).

Now we are ready to derive the discrete version of (2–44). Using the equations derived above, we obtain

$$\hat{h} = \begin{cases} \dfrac{\displaystyle\sum_{k=0}^{N/2} r_k^2 k\,\mathrm{sinc}(\dfrac{2\pi k}{N})\sin 2\pi k h}{2\pi\displaystyle\sum_{k=0}^{N/2} r_k^2 k^2\,\mathrm{sinc}(\dfrac{2\pi k}{N})^2} & \text{(forward)}, \\[4em] \dfrac{\displaystyle\sum_{k=0}^{N/2} r_k^2 k\,\rho(\dfrac{2\pi k}{N})\cos(2\pi k h + \alpha(\dfrac{2\pi k}{N}))}{2\pi\displaystyle\sum_{k=0}^{N/2} r_k^2 k^2\,\rho(\dfrac{2\pi k}{N})^2} & \text{(centered)}. \end{cases}$$

A comparison of these results with the continuous case in equation (2–44) shows that they are very similar. The primary feature of the result, the sine term in the numerator, is retained (although its phase is shifted in the centered case); and the contribution of each term is modified by the sinc function in the forward case and the ρ function in the centered case. In either case, as N tends to infinity, the discrete result tends to the continuous result, as would be expected.

Symmetric version.   The symmetric version of (2–33) is

$$\hat{h} = \frac{\sum_x (I_1(x) - I_2(x))(I_1'(x) + I_2'(x))}{\sum_x I_1'(x)^2 + I_2'(x)^2}.$$

Equivalently, expressed using inner products (and dropping the arguments),

$$\hat{h} = \frac{\langle I_1 - I_2, I_2' + I_1'\rangle}{\langle I_2', I_2'\rangle + \langle I_1', I_1'\rangle}.$$

Expanding, we obtain

$$\hat{h} = \frac{\langle I_1, I_2' \rangle + \langle I_1, I_1' \rangle - \langle I_2, I_2' \rangle - \langle I_2, I_1' \rangle}{\langle I_1', I_1' \rangle + \langle I_2', I_2' \rangle}.$$

But I_1 and I_2 differ only in phase, so as a consequence of equation (2–42), $\langle I_1, I_1' \rangle = \langle I_2, I_2' \rangle$ and $\langle I_1', I_1' \rangle = \langle I_2', I_2' \rangle$. Furthermore, by symmetry $\langle I_1, I_2' \rangle = - \langle I_2, I_1' \rangle$. Thus

$$\hat{h} = \frac{2 \langle I_1, I_2' \rangle}{2 \langle I_2', I_2' \rangle}.$$

But this is the same as equation (2–43), so the symmetric version yields the same result as the unsymmetric version! This is of course partly a consequence of the rather idealized view of I_1 and I_2, viz. that they are periodic and that $I_1(x)$ is in fact exactly the same as $I_2(x)$ shifted on the x axis.

Consequences with respect to smoothing and iteration.   This section analyzes the effects of smoothing and iteration given these results. We use the continuous equation (2–44), with the understanding that the results only apply approximately. Let us analyze the case where $I_1(x) = I_2(x + h)$; the goal is to calculate h.

Let us examine the iterative scheme for this one-parameter case in detail. We start with an initial estimate $\hat{h}_0$ of the disparity h. At each iteration the goal is to improve the current estimate $\hat{h}_i$. Ideally, one would like to compute the error between the current disparity estimate and the actual disparity, defined to be

$$e_i = h - \hat{h}_i;$$

this would allow the exact computation of h. But as described earlier in this chapter, computing e_i is itself a disparity estimation problem, because

$$h - \hat{h}_i = \text{disparity between } I_1 = I_2(x + h) \text{ and } I_2(x + \hat{h}_i).$$

The solution is to use the method of differences to compute this disparity; but in fact the method of differences computes only an estimate of the disparity, say $\hat{e}_i$. Thus,

$$h = \hat{h}_i + e_i, \quad \text{but}$$

$$\hat{h}_{i+1} = \hat{h}_i + \hat{e}_i. \tag{2–47}$$

Equation (2–47) is the basis for iteration in the method of differences. In summary, computing the improvement $\hat{e}_i$ to a disparity estimate $\hat{h}_i$ is itself a disparity estimation problem.

Now, according to equation (2–44), given a disparity of $e_i = h - \hat{h}_i$, the method of differences computes a disparity estimate $\hat{e}_i$ given by

$$\hat{e}_i = \frac{\sum\limits_{k \geq 0} r_k^2 k \sin 2\pi k e_i}{2\pi \sum\limits_{k \geq 0} r_k^2 k^2} = \frac{\sum\limits_{k \geq 0} r_k^2 k \sin 2\pi k (h - \hat{h}_i)}{2\pi \sum\limits_{k \geq 0} r_k^2 k^2}. \tag{2–48}$$
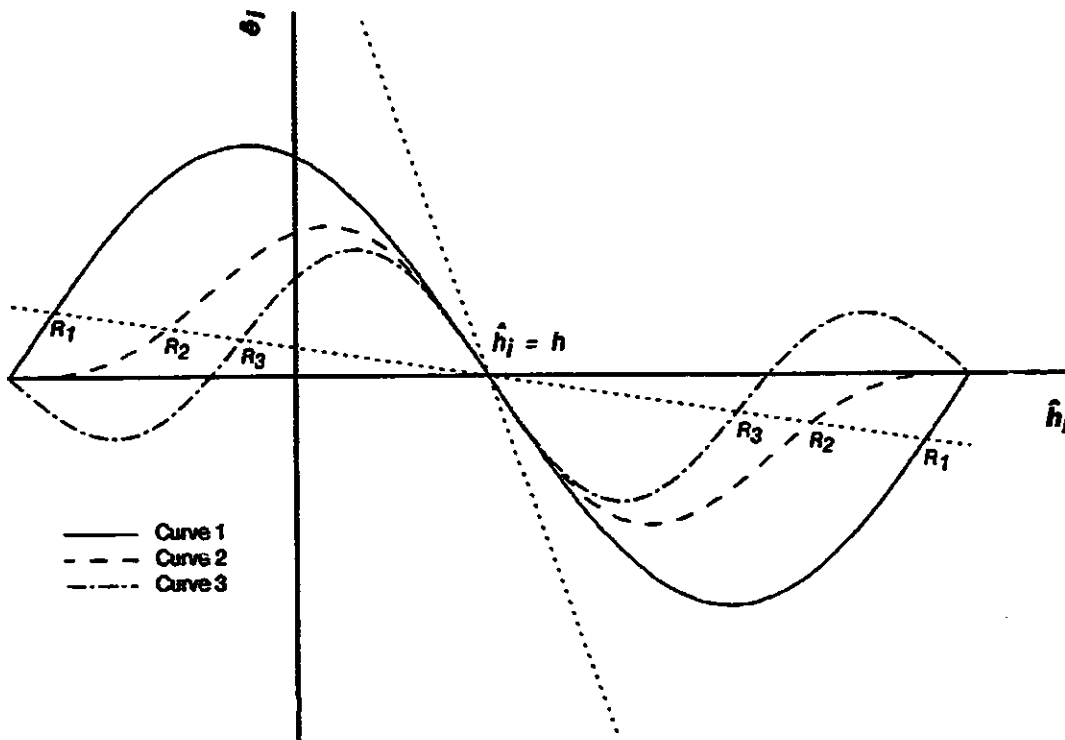
Figure 2-2. Horizontal axis shows ith disparity estimate $\hat{h}_i$ of the actual disparity h. Vertical axis shows resulting estimate $\hat{e}_i$ of the disparity error $h - \hat{h}_i$, as computed by the method of differences. Curves are examples of error estimates $\hat{e}_i$ for various assumptions about image power spectrum r_k: curve 1 corresponds to $r_1 = 1$, $r_2 = 0$; curve 2 to $r_1 = 1$, $r_2 = \frac{1}{2}$; curve 3 to $r_1 = 1$, $r_2 = 1$. All three curves cross the horizontal axis where the error estimate is zero, at $\hat{h}_i = h$, and have slope -1 at that point. Dotted lines have slope -0.1 and -1.9, corresponding to $c = 0.9$. R_1, R_2 and R_3 indicate the points where the curves cross the dotted line, which are the limits of the region of convergence for each curve, relative to the given c.

As discussed earlier, the power spectrum of a typical image is dominated by low frequency terms, and r_k^2 falls off rapidly as k increases. This is particularly true in smoothed images because smoothing attenuates the high-frequency terms. Therefore, as a first-order approximation, let us consider only the first two terms of (2-48), corresponding to $k = 1$ and $k = 2$. Three such cases are graphed in Figure 2-2. In the simplest case, curve 1, the second term $r_2 = 0$, so the curve is just a sine wave. Even though it ignores most of the power spectrum, experiments will show that the first term reveals the general form of the error estimate $\hat{e}_i$.

Nevertheless, the effect of the higher-frequency terms, as illustrated by curves 2 and 3, is important. However, it is necessary to understand the convergence behavior of equation (2-48) before the effect of the higher-frequency terms can be understood.

If the current estimate $\hat{h}_i$ is equal to the actual disparity h (corresponding to the point where the curves cross the horizontal axis), then the computed error $\hat{e}_i$ will be 0 according to equation (2-48), and so all subsequent estimates $\hat{h}_i$ will be equal to h as well. Thus the actual parameter value h is in fact a convergence point of the algorithm. Now, for any region R, an estimate $\hat{h}_i$ in that region will converge to the correct value h if there is a constant $c < 1$ (independent of $\hat{h}_i$) such that

$$\text{if} \quad \hat{h} \in R \quad \text{then} \quad -c|\hat{h}_i - h| - (\hat{h}_i - h) \le \hat{e}_i \le c|\hat{h}_i - h| - (\hat{h}_i - h) \qquad (2\text{-}49)$$

$$\text{and} \quad \hat{h}_i + \hat{e}_i \in R. \qquad (2\text{-}50)$$

This is because if $\hat{h}_i \in R$ then

$$|\hat{h}_{i+j} - h| \le c^j |\hat{h}_i - h| \to 0,$$

and so $\hat{h}_i \to h$. The smaller c, the faster the convergence.

Condition (2-49) can be represented graphically by two lines crossing the horizontal axis at the same point h that the graph of $\hat{e}_i$ does, as shown by the dotted lines in Figure 2-2 for $c = 0.9$. The convergence region R must be contained in the interval between the points where the graph of the error estimate $\hat{e}_i$ leaves the area between the two lines. These limits are indicated by R_1, R_2 and R_3 for the three curves in Figure 2-2. Equation (2-50) must also be satisfied. One sufficient condition is that the region R be an interval that is symmetric about h. Another sufficient condition is that the graph of the error estimate $\hat{e}_i$ does not leave the area between the horizontal axis and the line of slope -1; this guarantees that the estimates $\hat{h}_i$ will always remain on the same side of h that $\hat{h}_0$ started on. Neither condition is necessary.

It remains to determine what value of c to use. As c goes to 1, the region R expands to include almost the entire interval between points where the error estimate $\hat{e}_i$ falls back to 0 (or exceeds twice the actual error); but the rate of convergence of the newly admitted points becomes slower and slower. On the other hand, the convergence estimate provided by $|h - \hat{h}_i| < c^i |h - \hat{h}_0|$ is rather pessimistic, particularly for points near the edges of the region R, because these points are quickly carried to parts of R where the convergence is much faster. The following reasoning demonstrates this: consider a point $\hat{h}$ at the edge of the region R for a given c; the first iteration carries it $1/(1 - c)$ of the way towards h; but, each successive error estimate is at least as large as the first (until the very end), so convergence will be obtained after about $1/(1 - c)$ iterations. This can be summarized as follows: each c between 0 and 1 determines a region R such that initial estimates within that region converge after about $1/(1 - c)$ iterations. The larger c the larger the region but the slower the convergence may be.

Now let us consider what happens as we add the higher order terms to the series of equation (2-48). The effect of adding the second term is illustrated by curves 2 and 3 of Figure 2-2. Higher-order terms correspond to larger values of k in equation (2-48), and thus add sine waves whose frequency is a multiple of the frequency of the first term. However, the amplitudes of these higher-order terms are in general smaller than the amplitude of the first term, and indeed fall off rapidly with increasing k. This is a consequence of the empirical observations concerning image spectra mentioned earlier. All of the sine waves cross the horizontal axis at $\hat{h}_i = h$, so that the error estimate at this point $\hat{e}_i$ is guaranteed to be zero. Near this point, all of the sine waves are going in the same direction, and so contribute with the right sign to the error estimate; their combined effect is normalized by the denominator of (2-48). Indeed, that equation guarantees that the error estimate $\hat{e}_i$ will have a slope of -1 at h, so that the error estimate will approximate the ideal straight line $\hat{e}_i = h - \hat{h}_i$ near h no matter what the power r_k^2 of each term of the Fourier series.

However, as the error becomes larger, that is as $\hat{h}_i$ moves away from h, some of the higher-order terms start to have the wrong sign. Near where the first term is at its maximum or minimum, the higher-order terms will have a harder time reducing the contribution of the first term to zero; but near the edge of the range of convergence, where the first order term has fallen back to zero, the higher-order terms can more easily reduce the estimate towards zero. The result is that the estimate including the higher-order terms will fall back to zero sooner than the estimate including only the first term. Thus, the net effect of adding higher order terms will be to reduce the range of convergence. Consider for example an extreme case: suppose the first term is absent and only the second term is present; then the convergence range will be half what it would be with only the first term. The curves in Figure 2-2 illustrate the point. In each case $r_1 = 1$; as r_2 is increased from 0 to $\frac{1}{2}$ to 1, the curve is primarily affected near the edge of the convergence region, and is affected in such a way as to decrease the region of convergence.

This analysis also leaves us in a better position to understand what smoothing the images accomplishes. The spatial-domain operation of smoothing can also be interpreted in the frequency domain as modifying the coefficients r_k^2 of the power spectrum. For linear smoothing, as for any linear filtering, the effect is to multiply each coefficient by a constant; in the case of smoothing, the constant is smaller for the higher-order terms. Thus the net effect of smoothing is to increase the range of convergence by reducing the effect of the higher-order terms.

The foregoing discussion is based on an analysis of the method of differences that makes some approximating assumptions. Even so, it provides considerable insight into what is happening in a real application. The reader is invited to examine the graphs of the error estimates from some experiments with real images, starting with Figure 4-11 on page 83. Even though these error estimates were made using images that were not in fact identical over a sparse set of points (so that the inner product calculations do not apply), the form of these graphs and their behavior under various degrees of smoothing follows the above discussion rather well. In particular, these graphs display a roughly sinusoidal shape,

with the presence of higher-frequency sine waves evident in many cases; and with smaller degrees of smoothing, the wavelength of the sine wave is shorter so the range of convergence is smaller.

2.7. Summary

The method of differences has been used in rudimentary form for motion detection and optical flow problems, as reported in the introduction. This chapter has extended the method to a wide variety of problems, and provided a theoretical basis for understanding its operation. This section summarizes the contributions of this chapter.

The method of differences characterizes matching problems as parameter estimation problems. This allows any problem in which one image has been transformed into another under the control of a few local parameters or a field of global parameters to be cast as an image matching problem. The basis of the technique is that each point in the image provides a linear constraint on the parameters. A least squares technique combines these constraints to yield a parameter estimate. This chapter has shown how the method of differences can be applied to one-dimensional registration, two-dimensional registration, photometric parameter estimation, affine registration, and mixed local and global parameter estimation. It has been shown how the method can be applied to problems in which a field of local transformation parameters is to be estimated. These examples are chosen to show the range of applicability of the technique, and to demonstrate how to apply the method to other problems.

Essential to the technique are iteration and smoothing. Iteration is necessary because the result produced is an estimate. Smoothing acts to increase the range of convergence. Also discussed are weighting techniques that could give more importance to those points judged more likely to provide good information about the parameters.

A Fourier analysis approach reveals that a relationship exists between the the power spectrum of the image and the results produced by the method of differences. Because the information in an image tends to be contained in the phase spectrum while power spectra tend to be very similar, this allows us to make some fairly general observations about the behavior of the algorithm, at least in the one-parameter case. In particular, estimates of the region of convergence and the number of iterations to achieve convergence are possible, and an understanding of the effect of smoothing on the algorithm is gained. Even though these observations are only approximations, experiments (reported in later chapters) show that the results obtained are similar to the predictions.

The next two chapters will take a detailed look at the theory involved in camera parameter estimation for optical navigation, and at some experiments to determine the performance. Then the subsequent two chapters will apply the method of differences to the determination of stereo depth maps.

Chapter 3

Optical Navigation: Theory

3.1. Introduction

This chapter demonstrates how the method of differences for image matching can be used for navigation. Navigation is a problem of determining the position and orientation of an object, such as an autonomous roving vehicle or a robot arm. By solving for six camera parameters the position and orientation of a camera, and thus of the object on which it is mounted, can be obtained. More specifically, suppose we are given a *reference image* I_1, a set P of *reference points* p in the reference image, and a set of distance values $z(p)$ at each reference point. Then, given a *test image* I_2, the method of differences can be used to find the position and orientation of the camera which produced the test image.

It would be desirable if the distances $z(p)$ of the reference points p did not have to be known a priori, as this would make the algorithm more useful. Unfortunately, the distances are necessary for determining the camera model. The intuition is that some indication of scale is necessary to distinguish between, say, looking at a room scene from a distance or looking at a model of the scene from close up. The distances to the reference points provide the indication of scale necessary to determine whether motion of the camera is large, as in the former case, or small, as in the latter. However, in many applications determining the distance values is not too much of a burden. For example, in the robot arm case the reference distances $z(p)$ can be determined once and for all using stereo.

The chapter begins by providing some motivation and background for the problem by describing some applications of optical navigation. Then the technique itself is discussed in detail. This involves defining the match parameters, which are the camera parameters, and developing the equations for them. Solving these equations involves computing and inverting a matrix; the implementation of these operations is discussed. Of interest in inverting the matrix is the conditions under which that operation is numerically stable. Geometric conditions for numerical stability on the three-space distribution of the reference points are derived. These conditions apply to any match-based navigation technique. Finally, methods

for obtaining reference points and their distance values are discussed. The next chapter will present some experimental results using these techniques.

3.2. Applications

Many robotic tasks require a knowledge of the position and orientation of the robot. This is because mechanical imperfections and environmental uncertainty make it impossible to know exactly how a robot will move in response to the commands sent to it and exactly what it will encounter in its surroundings. Optical navigation takes its place alongside various types of mechanical navigation and global positioning systems to correct this problem, and indeed has distinct advantages with respect to responding to environmental uncertainty over those methods. Such tasks can be classified along several dimensions; three of these are the nature of the robotic agent, the nature of the environment, and the manner in which the optical navigation is used.

Robotic agents. For our purposes, robotic agents can be roughly divided into two classes: fixed-base robot arms, and autonomous roving vehicles (although many vehicle designs call for the rover to have a robotic arm or manipulator of some sort). In both cases optical navigation can play an important role, although the nature of the navigation is different.

For example the, world in which a manipulator moves is generally smaller than that in which a rover moves, and so the nature and quality of the image obtained may differ (consider such effects as depth of field). Moreover, the two types of agent differ with respect to the availability of other sources of information about the robot's movement. A rover might well be capable of inertial or radio navigation in addition to optical navigation. On the other hand robot arms typically operate in environments where such techniques as structured lighting may feasibly provide additional information, while robotic vehicles are likely to be required to operate in a more unconstrained environment. Such additional sources of knowledge could be incorporated into the method, for example to provide the necessary initial estimate of position.

A second point of difference is that the mechanisms for control of an arm and a rover are quite different: an arm is generally controlled by the coordinated rotations of a number of joints, while the a rover is moved under the power of a number of wheels or legs. Thus, the method must be adapted differently in each case if it is desired to directly solve for the joint positions, the wheel rotations, or even the control signals that cause those movements.

Another point of difference is in the degree of constraint that exists in the motions of the robot. Typically, a manipulator will be able to move in all six degrees of freedom, while a rover may only have freedom in say the pan, x, and z motions. These constraints bear on the specific formulations of the technique for the different tasks.

Environment. In a known environment, the robot must perform a series of maneuvers with respect to a set of objects in the environment whose nature and approximate position

are known in advance. In an unknown environment the robot must be prepared to encounter anything, or at least a variety of objects. Tasks in which a robot arm must operate in an unknown environment are conceivable, but unlikely; therefore the discussion will be confined to the rover case, but the remarks concerning the rover operating in a known environment will apply equally well to the robot arm case.

In the case of a known environment, the scenario for the use of the method of differences is as follows: the robot is taken through the series of operations that it will be expected to perform. As it does this it records a number of camera views sufficient to cover the substantially different situations the robot will encounter. These images will serve as landmarks with respect to which the robot will later navigate. Then a number of reference points (a few to several dozen) are chosen from each image and are assigned a distance from the camera. This can be done a number of ways: a second camera in a known position with respect to the first can provide a stereo baseline for making the measurements; a known model of the environment can be fitted to those points; the distances can be directly measured; the distances can be obtained by structured lighting; and so on. In any case, since this is a training step to be done only once, the assignment of distances need not be completely automated. This concludes the training process. Then, at each step of an actual run, the image received by the camera will be compared against one of the stored reference images (actually, only the positions and intensities of the reference points need to be stored), and the method of differences will be used to determine the position of the robot relative to the reference coordinate system. In a variation of this process, the method of differences can be used to directly solve for the control signals to be fed to the robot.

The case of an unknown environment is more difficult. It is not possible to store reference images, so the process of selecting and determining the distances of reference points must be carried out anew each time the rover encounters a new situation. Techniques for doing this, such as by automatic stereo vision, are themselves the subject of research. Once the reference points are located and their distances determined, they can be used to navigate until they are lost from view. If necessary, the stereo solver can be called again at each step to refine the estimates of the distances and to determine the distances of new reference points acquired to replace old ones that have disappeared from view.

Manner of application. Optical navigation can be used in a robot in one of two ways. If the position of the robot can be calculated quickly enough (for example with the aid of special-purpose hardware), then the result can be used in a continuous feedback loop. Two aspects of the method are favorable for this mode: in a feedback loop the range of convergence of the algorithm need not be large, because the position will be calculated frequently enough that the robot will have moved relatively little during that time. Moreover, the method need not accurately compute the position of the robot, but rather needs to be taught what the camera should see when it is in the desired position; as long as the method is able to provide signals to move in the proper direction when the camera is out of position and provided that the method can detect when the camera is seeing what it is expected see,

the position of the camera will converge to the correct position. Another way of expressing this is that the position coordinates reported by the method need not be accurate world coordinates, but rather can be some distorted set of private coordinates. All that is needed is that these private coordinates have a one-to-one and consistent relationship with world coordinates, and that the robot has learned its movements in these private coordinates, so that it can repeat them accurately. Of course proper precautions are necessary to prevent oscillation.

Even if the method cannot be applied fast enough to be used in a feedback mode, it can still be fruitfully used in a "stop-and-go" mode. In such a case the rover will move greater distances between each application of the method, and so the demands on the optical navigation are greater: it must exhibit in this case a greater range over which it will converge to the proper value.

Scenarios. Optical navigation has applications in autonomous roving vehicles and manufacturing arms, among others. Let us now consider the scenarios under which a navigation technique like the one described in this chapter could be used.

An autonomous rover moving in the real world requires a navigation system. An optical navigation technique is particularly desirable because of its adaptability to a wide variety of situations. In the scenario envisioned here, the rover is equipped with a pair of cameras which enables it to determine the nature of its environment, using for example the techniques developed by Moravec (1980). One image of the stereo pair serves as the reference image, and the rover is then moved to a new location. The position of the rover at this new location is determined using one of the pictures at the new location as the test image in the optical navigation technique described here.

As for the manufacturing application, consider the following scenario: a sequence of parts move past a robot arm on a conveyor belts. As each one passes the arm the arm must perform some task on it requiring knowledge of the part's position and orientation, such as inserting a screw. There are two problems here: acquiring the position of the part relative to the arm, and guiding the arm into proximity with the part where it can perform its task. It is assumed that a reference image of the object is prepared, and that its three-dimensional form and the path of approach to the object can be obtained by training runs, as discussed at the end of this chapter. Then, with a camera mounted on the arm providing test images, both the position and the guidance problems can be solved by an optical navigation system.

If the feedback is fast enough, guiding the arm can be done by the type of "optical servoing" discussed above. The assumption is that the mechanical guidance of the arm is not sufficiently accurate and the visual feedback can be used to keep it on the proper course. Perhaps more to the point, if optical servoing could be made sufficiently fast and accurate, robot arms would not need accurate mechanical guidance.

The use of the method of differences imposes two constraints. First, the reference picture must not be too different from the test picture; for example, in the manufacturing

scenario, if the reference picture recorded the part from the front but it was seen from the back in the test image, then neither the method of differences nor any optical navigation technique could work. The second constraint is that a sufficiently accurate estimate of the position of the test camera must be available. In the rover case, this is satisfied if the rover is mechanically constructed so that it can be moved with reasonable accuracy; in the manufacturing case, the part must be placed on the conveyor belt within some tolerance. As the experiments of the next chapter will show, the tolerance is not excessively restrictive. However, if either of these constraints is violated, all is not lost. Violation of the first constraint would require maintaining several reference images and using the one nearest to the current position; violation of the second constraint would require searching in position space, albeit at coarse resolution determined by the range of convergence of the method.

The remainder of this chapter discusses the technique itself, its implementation, and its numerical stability. Finally, it is shown how the reference image, the reference points, and their distances can be obtained.

3.3. The camera model

A camera model is a mathematical model of the image formation process of a camera. A camera model consists of two parts: a geometric part and a photometric part. The geometric part expresses the position and orientation of the camera relative to a reference coordinate system, and consists of six parameters: three for the position and three for the orientation. The photometric part expresses the relationship between the intensity of light reflected from a point in three space and the intensity value reported for that point by the camera. It is modeled here by a simple linear transformation expressing relative bias and gain.

The camera model described here is a simple one; additional parameters could be used. For example, non-linear geometric distortions introduced by the optics and by the scanning of the camera retina could be modeled perhaps by polynomials (Moravec, 1980). The coefficients of these polynomials would be additional match parameters to be solved for. This was not done here because geometric distortions in the experimental data (provided by Moravec) had already been removed. Moreover, in a real application these distortions would be constants of the sensors that could be compensated for in advance. As another example, the photometric relationship between the sensors could be modeled by something more complex than the simple linear relationship used. This was not done because even the simple linear relationship between intensity values was not needed; that is, the cameras were assumed to report identical intensity values for corresponding scene points. This is of course not true in actuality both because of sensor differences and because of such photometric effects as specular reflection. Nevertheless, the photometric parameters could be ignored because the reference points were chosen to be near edges, where photometric differences between the images have relatively little effect on the derived match position and thus relatively little effect on the match parameters. It is an open question whether using the photometric parameters could improve the accuracy.

Moreover, the camera model described here is limited in that it also assumes only two cameras. Presumably, multiple cameras with known geometric relationships could provide additional information. In particular, additional cameras would reduce noise, reduce photometric error, and reduce the possibility of the problem being numerically unstable due to the geometry of the scene. However, these conjectures are not explored in this thesis.

Let us now proceed to the geometric part of the camera model. Every camera determines a coordinate system. The origin is at the center of projection of the camera (the pinhole in a pinhole camera); we will take the positive x direction to be to the right along the scan lines, the positive y direction to be up perpendicular to the scan lines, and the positive z direction to be forward, perpendicular to the image plane. This is illustrated in Figure 3-1. The model includes two cameras: a reference camera (camera 1) and a test camera (camera 2). The coordinate system of the reference camera will be the world coordinate system, so that the position of the test camera is given as a vector r between the origins of the camera coordinate systems, expressed in the reference coordinate system. The position vector r can be specified by its three components, r_x, r_y, r_z (using a rectilinear coordinate system), or by the azimuth α_{AZ}, elevation α_{EL}, and range r (using a polar coordinate system). The two systems are related by

$$r = [r_x \quad r_y \quad r_z] \tag{3-1}$$

$$= [0 \quad 0 \quad r] A_{EL} A_{AZ}, \tag{3-2}$$

where A_{EL} and A_{AZ} are the matrices which rotate by the elevation and azimuth angles respectively (see Appendix A).

Assume as given a reference image, I_1, and suppose that we know at several points p of this image the z coordinate $z(p)$ of the corresponding three-space point. (These z values are not part of the camera model, but must be given a priori). Then the corresponding three-space point q is given by

$$q = \left[\frac{p_x z(p)}{f} \quad \frac{p_y z(p)}{f} \quad z(p) \right], \tag{3-3}$$

where f is the focal length of the reference camera. This equation is of course the inverse of the projection equation.

Now suppose the test camera is related to the reference coordinate system by the position vector r (which has three parameters, as in (3-1) or (3-2)), and by the orientation parameters pan (α_{PA}), tilt (α_{TI}), and roll (α_{RO}). Then the three-space point q in the reference coordinate system is u in the test coordinate system, given by

$$u = (q - r) A_{PA} A_{TI} A_{RO}, \tag{3-4}$$

where A_{PA}, A_{TI}, and A_{RO} are the rotation matrices for pan, tilt, and roll respectively, as given in Appendix A.
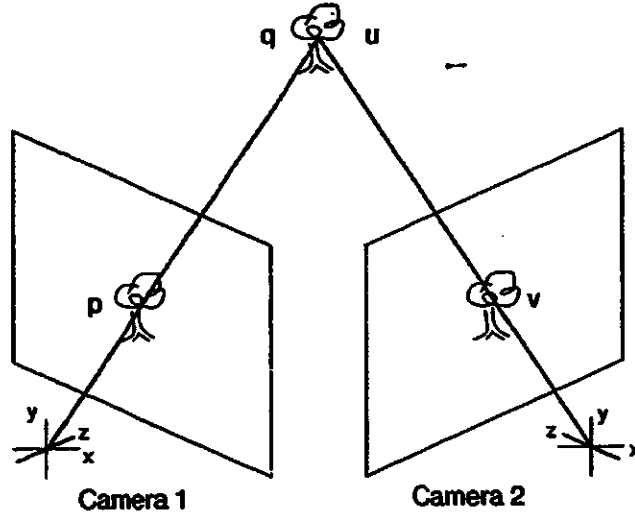
Figure 3-1. Camera model. Camera 1 defines the reference picture and coordinate system, with the origin at the "pinhole." For any point $p = [p_x \quad p_y]$ in the image there is a point $q = [q_x \quad q_y \quad q_z]$ in three-space which produced the image, at depth $z(p) = q_z$. This point is expressed as $u = [u_x \quad u_y \quad u_z]$ in the Camera 2, or test, coordinate system. The relationship between q and u is a function parameterized by the six camera parameters: three for the relative positions of the cameras and three for their relative orientation. Finally, the three-space point appears at the point $v = [v_x \quad v_y]$ in the Camera 2 image plane. The points p and v are said to correspond.

Finally, the point v on the test image plane that corresponds to u is given by the projection equation:

$$v = \left[\frac{u_x f}{u_z} \quad \frac{u_y f}{u_z} \right]. \tag{3-5}$$

Equations (3-3), (3-4), and (3-5) allow us to compute a point v on the test image that corresponds to a given point p with depth z on the reference image, given the six geometric parameters.

As mentioned above, the photometric model assumed in this development is such that the intensity values of the two images are linearly related by two parameters: β (bias, brightness) and γ (gain, contrast). Thus, the intensity values I_1 and I_2 from the two cameras respectively for corresponding points are related by

$$I_2 = \gamma I_1 + \beta \tag{3-6}$$

These parameters are included in the model only for completeness. It is not known whether these parameters contribute to the accuracy of the other parameters (which are the ones of real interest), since the photometric parameters were ignored in the experiments.

3.4. Solving for the camera parameters

From equation (3-6) one would expect

$$0 = I_2(\mathbf{v}) - \gamma I_1(\mathbf{p}) - \beta,$$

where $\mathbf{p}$ and $\mathbf{v}$ are corresponding points on the reference and test images respectively, as defined above. For any given estimate of the geometric and photometric parameters, define the amount by which the right hand side of the above equation differs from zero as the error $E_\mathbf{p}$ associated with the point $\mathbf{p}$ in the reference image:

$$E_\mathbf{p} = I_2(\mathbf{v}) - \gamma I_1(\mathbf{p}) - \beta. \tag{3-7}$$

Note that this equation explicitly shows that $E_\mathbf{p}$ depends on the two photometric parameters, but it also of course depends on the six geometric parameters because of the relationship between $\mathbf{v}$ and $\mathbf{p}$ set forth above.

This error is presumably due to misestimates of the geometric and photometric parameters, so it is desired to adjust these to minimize the error. This cannot be done on the basis of one point $\mathbf{p}$ alone, because each point $\mathbf{p}$ provides only one linear constraint on the eight parameters. Thus it is necessary to combine the information from at least as many points as there are parameters. Therefore let us assume that we have a set P of points $\mathbf{p}$ and the distance information $z(\mathbf{p})$ at each point. As mentioned at the beginning of the chapter, the distances are needed to determine the scale of the motion, r. Then let us define the total error E associated with the set P to be

$$E = \sum_{\mathbf{p} \in P} E_\mathbf{p}^2.$$

This is a non-linear function of the quantities we wish to minimize with respect to, namely the six geometric and two photometric parameters. By making a linear approximation, as detailed below, we can make $E_\mathbf{p}$ linear in the (change in) the parameters, thus making E quadratic, and so again getting a linear system of equations to solve when we differentiate E with respect to each of the parameters and set equal to zero. This change in the parameters is suitable for modifying the estimate of the camera parameters as an iteration in an iterative scheme.

The linear approximation is as follows. Let $\mathbf{c}$ denote the vector of camera parameters, namely

$$\mathbf{c} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 \end{bmatrix}$$

$$= \begin{bmatrix} r_x & r_y & r_z & \alpha_{PA} & \alpha_{TI} & \alpha_{RO} & \beta & \gamma \end{bmatrix}.$$

Suppose we have an estimate of these camera parameters. If we adjust the estimate to $c + \Delta c$, each error estimate E_p changes, to a linear approximation, to $E_p + \Delta c(D_c E_p)$, and so the total error estimate is

$$E \approx \sum_{p \in P} (E_p + \Delta c(D_c E_p))^2.$$

Here $D_c E_p$ is of course just the vector of partial derivatives of the error E_p associated with the point p in the reference image with respect to each of the eight camera parameters. It will be shown how to compute $D_c E_p$ below. Now, we wish to minimize E with respect to Δc, so let us set

$$0 = D_{\Delta c} E$$

$$\approx D_{\Delta c} \sum_{p \in P} \left(E_p + \Delta c(D_c E_p)\right)^2$$

$$= \sum_{p \in P} D_{\Delta c} \left(E_p + \Delta c(D_c E_p)\right)^2$$

$$= 2 \sum_{p \in P} E_p(D_c E_p) + \Delta c(D_c E_p)(D_c E_p)^{\mathsf{T}},$$

so that the Δc which minimizes E (to a linear approximation) is

$$\Delta c = \left(- \sum_{p \in P} E_p(D_c E_p)\right) \left(\sum_{p \in P} (D_c E_p)(D_c E_p)^{\mathsf{T}}\right)^{-1}.$$

The structure of this can be made clearer by showing the elements of the matrix explicitly:

$$\Delta c = \begin{bmatrix} -\Sigma_1 \\ -\Sigma_2 \\ \vdots \\ -\Sigma_8 \end{bmatrix} \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{18} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{28} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{81} & \Sigma_{82} & \cdots & \Sigma_{88} \end{bmatrix}^{-1}, \tag{3-8}$$

where

$$\Sigma_i = \sum_{p \in P} E_p(D_{c_i} E_p), \quad \text{and} \tag{3-9}$$

$$\Sigma_{ij} = \sum_{p \in P} (D_{c_i} E_p)(D_{c_j} E_p). \tag{3-10}$$

This matrix will be discussed later in more detail. The numerical stability of inverting it will be of particular concern.

Now the discussion turns to computing $D_c E_p$. First let us define the vector g of geometric camera parameters:

$$\mathbf{g} = \begin{bmatrix} \mathbf{r}_x & \mathbf{r}_y & \mathbf{r}_z & \alpha_{PA} & \alpha_{TI} & \alpha_{RO} \end{bmatrix}.$$

Now, referring to the definition of E_p in equation (3-7), compute $D_c E_p$ as follows:

$$D_c E_p = \begin{bmatrix} D_g E_p \\ D_\beta E_p \\ D_\gamma E_p \end{bmatrix} = \begin{bmatrix} (D_g \mathbf{u})(D_u \mathbf{v})(D_v I_2) \\ -1 \\ -I_2(\mathbf{p}) \end{bmatrix}. \tag{3-11}$$

The product of matrices $(D_g \mathbf{u})(D_u \mathbf{v})(D_v I_2)$ is of course just due to the chain rule; these matrices, written out explicitly, are

$$D_g \mathbf{u} = \begin{bmatrix} D_{r_x}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \\ D_{r_y}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \\ D_{r_z}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \\ D_{\alpha_{PA}}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \\ D_{\alpha_{TI}}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \\ D_{\alpha_{RO}}(\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A_{RO} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} A_{PA}A_{TI}A_{RO} \\ \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} A_{PA}A_{TI}A_{RO} \\ \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} A_{PA}A_{TI}A_{RO} \\ (\mathbf{q}-\mathbf{r})A'_{PA}A_{TI}A_{RO} \\ (\mathbf{q}-\mathbf{r})A_{PA}A'_{TI}A_{RO} \\ (\mathbf{q}-\mathbf{r})A_{PA}A_{TI}A'_{RO} \end{bmatrix}; \tag{3-12}$$

$$D_u \mathbf{v} = \begin{bmatrix} D_{u_x}\begin{bmatrix} \dfrac{u_x f}{u_z} & \dfrac{u_y f}{u_z} \end{bmatrix} \\ D_{u_y}\begin{bmatrix} \dfrac{u_x f}{u_z} & \dfrac{u_y f}{u_z} \end{bmatrix} \\ D_{u_z}\begin{bmatrix} \dfrac{u_x f}{u_z} & \dfrac{u_y f}{u_z} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \dfrac{f}{u_z} & 0 \\ 0 & \dfrac{f}{u_z} \\ \dfrac{-u_x f}{u_z^2} & \dfrac{-u_y f}{u_z^2} \end{bmatrix}; \tag{3-13}$$

$$D_v I_2 = \begin{bmatrix} x \text{ component of the gradient of the test image} \\ y \text{ component of the gradient of the test image} \end{bmatrix}.$$

Here A'_{PA} is the derivative of the rotation matrix A_{PA} with respect to α_{PA}, and similarly for A_{TI} and A_{RO}; see Appendix A for the full definition of these matrices. At this point the reader may want to verify that the above matrices are conformal and so the product $(D_g \mathbf{u})(D_u \mathbf{v})(D_v I_2)$ is well-defined.

Compute	From	Eqn.
r_x, r_y, r_z	$\alpha_{AZ}, \alpha_{EL}, r$	(3-2)
A_{PA}, A_{TI}, A_{RO}	$\alpha_{PA}, \alpha_{TI}, \alpha_{RO}$	Appendix A
$D_g u$	$r, \alpha_{PA}, \alpha_{TI}, \alpha_{RO}$	(3-12)

Table 3-2. Computing Δc. Operations outside the inner loop.

3.5. Computing the matrix

The purpose of obtaining Δc as in equation (3-8) is to use it in an iterative scheme. The iterations starts with an estimate of c, and at each step replaces c with $c + \Delta c$. Each iteration involves computing the matrix $[\Sigma_{ij}]$, and then inverting it to obtain Δc. These two steps are discussed in this and the following section.

Computing the matrix is straightforward using the above equations, although some care should be taken to avoid unnecessary computations. This section takes as given the reference image I_1 and test image I_2, an estimate of camera parameters c, and a set P of points p for which we have depth measurements $z(p)$. The goal is to compute an adjustment Δc to the camera parameters c to reduce the error E.

There are a number of computations which are common to each reference point p and so can be moved outside the inner loop. Efficiency is of no great concern in carrying out these computations, because they are done only once per iteration. These operations involve computing the components of r; A_{PA}, A_{TI}, A_{RO}; and $D_g u$. These quantities change on each iteration as the match parameters change. The necessary operations are summarized in Table 3-2.

Now, the vector $[\Sigma_i]$ and matrix $[\Sigma_{ij}]$ must be computed from equations (3-9) and (3-10) respectively. This requires computing several sums over all p in P, so efficiency is of considerable concern. These quantities are computed in a series of steps summarized in Table 3-3. Computing the gradient $D_v I_2$ is discussed in Appendix B. A considerable part of that computation can be completed outside the inner loop. The figure reported for the number of operations is a rough count of the number of additions and multiplications required for each step. These are fairly naïve figures in that some optimizations are not included. Nevertheless, the conclusion remains that on the order of 100 to 150 operations per reference point p per iteration are required. Thus this is by far the most computationally expensive part of the operation, and is worth some effort to optimize. For example, this would be the part of the computation to do in special-purpose hardware.

3.6. Inverting the matrix

Having obtained the matrix $[\Sigma_{ij}]$ it must now be inverted to solve the system of equations it represents. Since this is only done once per iteration of computing c efficiency

Compute	From	Eqn.	No. Opns.
q	p	(3-3)	4
u	q	(3-4)	18
v	u	(3-5)	4
$D_g u$	r, α_{PA}, α_{TI}, α_{RO}	(3-12)	36
E_p	I_1, I_2, v, γ, β	(3-7)	3
$D_v I_2$	I_2, v	Appendix B	8
$D_u v$	f, u	(3-13)	6
$D_c E_p$	$D_g u$, $D_u v$, $D_v I_2$, $I_2(p)$	(3-11)	35
Σ_i	$D_{c_i} E_p$, E_p	(3-9)	6
Σ_{ij}	$D_{c_i} E_p$, $D_{c_j} E_p$	(3-10)	21

Table 3-3. Computing Δc. Operations per point $p \in P$.

is of no great concern, but one is concerned about accuracy and the conditions under which inverting the matrix is numerically stable. This section explores the conditioning of this matrix and develops a stability measure based the geometry of the three-space points q corresponding to the reference points p, and independent of the photometry of the particular images. This measure is applicable to any match-based navigation technique. The next section explores the implications of this geometric stability measure.

There are two related indications of the stability of inverting a matrix. The first, the *condition number* of the matrix, is directly related to the accuracy of the result but is difficult to analyze. The condition number is dependent on a particular definition of a matrix norm, which is in turn dependent on a particular definition of a vector norm. Let $\|x\|$ denote some norm of vector x. Then the matrix norm $\|A\|$ subordinate to this vector norm is defined by

$$\|A\| = \max_x \frac{\|Ax\|}{\|x\|}.$$

This definition of matrix norm allows one to define the condition number $\kappa(A)$ of a matrix as

$$\kappa(A) = \|A\| \, \|A^{-1}\|. \tag{3-14}$$

The significance of the condition number is as follows: consider the problem of solving the linear system $Ax = b$. Suppose we perturb the elements of A by some amount ΔA; denote the error that this induces in the solution x by Δx. That is,

$$(A + \Delta A)(x + \Delta x) = b.$$

It can be shown that the relative error in the calculated solution is bounded by the condition number of the matrix times the relative error in the matrix; that is,

$$\frac{\|\Delta x\|}{\|x + \Delta x\|} \leq \kappa(A) \frac{\|\Delta A\|}{\|A\|}. \tag{3-15}$$

(See for example Dahlquist & Björck, 1974). Thus, a small condition number means the system is stable, and a large condition number means the system is unstable. Now, it can further be shown that in the case of the L_2 norm the condition number $\kappa_2(A)$ of a symmetric matrix A with eigenvalues λ_i is given by

$$\kappa_2(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}.$$

(See for example Pizer, 1975). The matrix $[\Sigma_{ij}]$ is indeed symmetric, so this equation applies.

However, the eigenvalues present certain analytic problems. (The numerical difficulty of computing the eigenvalues is of no concern because the matrices used here are relatively small and will be inverted infrequently). Therefore, in some cases the *determinant* will be used as an approximate indication of the stability of the matrix. That the determinant is related to the condition number can be seen in (3-14): the term $\|A^{-1}\|$ has a factor of $1/\det(A)$, so one would expect that when the determinant is small the condition number would be large and vice versa. Indeed, when the determinant is zero, the matrix is singular, which is the extreme case of instability. It will be seen that, in the matrices we are dealing with, the determinant correlates well with the condition number.

Both the condition number and the determinant suffer from a problem of the units in which the variables (the camera parameters c in this case) are expressed. Changing the units in which a variable is given is of course tantamount to scaling the variable. Scaling a variable c_k is equivalent to dividing both row k and column k of the matrix $[\Sigma_{ij}]$ by the scale factor, since $\Sigma_{ij} = \sum_{p \in P}(D_{c_i} E_p)(D_{c_j} E_p)$. This divides the determinant by the square of the scale factor. The effect on the condition number is more subtle, but it is nonetheless present. Therefore, for the purposes of computing the conditioning of the problem, let us replace our matrix $[\Sigma_{ij}]$ by a normalized matrix $[\overline{\Sigma}_{ij}]$, where

$$\overline{\Sigma}_{ii} = 1, \tag{3-16}$$

$$\overline{\Sigma}_{ij} = \frac{\sum_{p \in P}(D_{c_i} E_p)(D_{c_j} E_p)}{\sqrt{\sum_{p \in P}(D_{c_i} E_p)^2}\sqrt{\sum_{p \in P}(D_{c_j} E_p)^2}}. \tag{3-17}$$

This matrix remains unchanged under a scaling of variables, and thus so do its condition number and determinant.

Under what conditions will this matrix be stable? Equations (3-16) and (3-17) reveal that the matrix $[\overline{\Sigma}_{ij}]$ is in fact a normalized covariance matrix. Thus, $\overline{\Sigma}_{ij}$ measures the

extent to which the "random variables" $D_{c_i}E_p$ and $D_{c_j}E_p$ are (linearly) correlated. If they are highly correlated, then one cannot tell the difference between the effect on the error E of changing c_i and c_j, and so we cannot hope to be able to solve the equations. This is reflected in the matrix as follows: if all the parameters' effects are independent, then all of the off-diagonal elements are zero, and the condition number of the matrix is one, which is the best possible. On the other hand, if two of the parameters, say c_i and c_j, always have the same effect, then Σ_{ij} and Σ_{ji} will be both be one, and the matrix will be singular. The determinant and the condition number measure the extent to which this is true, but applied to all variables in combination rather than just by pairs.

It would be desirable to make some general statements about the geometric conditions under which our equations are stable and thus one can determine the camera position c. Unfortunately, the matrix $[\Sigma_{ij}]$ includes too much information. In particular, it includes information about the photometric properties of the object as well as the geometric properties. But our intuition tells us that there will be circumstances under which the geometry alone should indicate that the camera position can't be determined. For example, a small change in the x position r_x has nearly the same effect on the appearance of the object as a change in the pan angle α_{PA} regardless of the shading of the object. (A similar statement holds for the y position r_y and the tilt angle α_{TI}.) The observation to make here is that changing these parameters has nearly identical effects on the position of the point v that matches p. That is, x motion is difficult to distinguish from pan for all objects because $D_{r_x}v$ is highly correlated with $D_{\alpha_{PA}}v$. Thus we would like a measure of the correlation of the geometric variables g alone. This would allow us to determine the best that could be expected from the method assuming the shading of the objects cooperates, and leave it to experiment to show that it usually does. Indeed, the geometric stability measure is and indication of the best that can be done by any match-based method.

Considering only the part of the matrix $[\Sigma_{ij}]$ due to the geometric variables (that is, the first six columns and six rows), we have

$$[\Sigma_{ij}] = \sum_{p \in P} (D_g E_p)(D_g E_p)^T$$

$$= \sum_{p \in P} (D_g v)(D_v I_2)(D_v I_2)^T (D_g v)^T. \tag{3-18}$$

The factor of $(D_v I_2)(D_v I_2)^T$ can be viewed as a sort of weight matrix; division by its sum as accounted for by when $[\Sigma_{ij}]$ is normalized to form $[\overline{\Sigma}_{ij}]$. If we wish to ignore the intensity information, this is the place to do it. Deleting the factor of $(D_v I_2)(D_v I_2)^T$, from equation (3-18), define the *geometric covariance matrix*

$$[\Gamma_{ij}] = \sum_{p \in P} (D_g v)(D_g v)^T. \tag{3-19}$$

This is contrasted with $[\Sigma_{ij}]$, which might be called the *photometric covariance matrix*. As with $[\Sigma_{ij}]$, let us define the normalized version $[\overline{\Gamma}_{ij}]$ of this matrix. For completeness, here

are the definitions of the elements of these matrices:

$$\Gamma_{ij} = \sum_{p \in P}(D_{g_i}v)(D_{g_j}v)^\top,$$

$$\overline{\Gamma}_{ii} = 1,$$

$$\overline{\Gamma}_{ij} = \frac{\sum_{p \in P}(D_{g_i}v)(D_{g_j}v)^\top}{\sqrt{\sum_{p \in P}(D_{g_i}v)(D_{g_i}v)^\top}\sqrt{\sum_{p \in P}(D_{g_j}v)(D_{g_j}v)^\top}}.$$

Examination of $\overline{\Gamma}_{ij}$ reveals that it is in fact precisely a measure of the correlation between the purely geometric effects of changes in the geometric camera parameters g_i and g_j. This is what intuition says we want to measure. It is important to realize that this condition number is a measure of the conditioning of the problem and limits the stability of any algorithm for optical navigation, be it by the method of differences or by some other method.

3.7. Geometric considerations

We are now in a position to determine what geometric configurations of camera and object lead to stable systems of equations. The basic conclusion will be that the object must be relatively three-dimensional; that is, its range of z values must not be small relative to its distance from the camera. As a corollary, for a given object, nearby with a short focal length lens is good, while faraway with a long focal length lens is bad.

Consider first the limiting case of faraway objects and long focal length lenses, namely orthography. Consider a fixed set P of reference points p with distances $z(p)$. From a purely geometric standpoint the only function of the p and $z(p)$ is to define a set Q of three-space points q via equation (3-3). Thus we take these three-space points q as our starting point. Without loss of generality the current α_{PA}, α_{TI}, α_{RO}, r_x, r_y, r_z estimates can all be taken to be zero. If not, one can solve for the new camera parameter estimates relative to a coordinate system in which the current estimates are all zero, and then translate the result to the actual coordinate system, without substantially affecting the conditioning of the problem. Thus the condition number derived for the case where they are zero applies as well to other cases. Under orthography, contrary to equation (3-5), $v = [u_x \quad u_y]$. Thus, referring to (3-12),

$$D_g v = (D_g u)(D_u v) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ -q_z & 0 & q_x \\ 0 & -q_z & q_y \\ -q_y & q_x & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ -q_z & 0 \\ 0 & -q_z \\ -q_y & q_x \end{bmatrix}.$$

This satisfies one's intuition: changing r_x (first row of $D_g v$) has the same effect, up to a scale factor of the distance to the point, as changing the pan α_{PA} (fourth row), and similarly for r_y and tilt α_{TI} (second and fifth rows respectively). Changing r_z (third row) has no effect under orthography, and changing the roll α_{RO} (sixth row) rotates the image. Carrying out the multiplication of equation (3-19) gives the unnormalized geometric covariance matrix:

$$[\Gamma_{ij}] = \sum_{q \in Q} \begin{bmatrix} 1 & 0 & 0 & q_x & 0 & q_y \\ 0 & 1 & 0 & 0 & q_x & -q_x \\ 0 & 0 & 0 & 0 & 0 & 0 \\ q_x & 0 & 0 & q_x^2 & 0 & q_y q_x \\ 0 & q_x & 0 & 0 & q_x^2 & -q_x q_x \\ q_y & -q_x & 0 & q_y q_x & +q_x q_x & q_x^2 + q_y^2 \end{bmatrix}.$$

The first observation to make about this matrix is that it is singular because the row and column corresponding to r_z are all zero. This is because under orthography the z values are ignored, it is not possible to solve for r_z. This will not be a problem under perspective, provided we normalize the matrix. Since orthography is a limiting case of perspective, the r_z row and column will be small when the object is far away. This will cause the matrix to be ill-conditioned unless we normalize it. This is because the small matrix row represents a difference in scale between the variables and is removed by normalization. Therefore, in the remainder of this analysis the r_z row and column will be deleted, thus solving for only five remaining geometric parameters.

Now, let us make the simple case where the points q in Q are symmetrically placed about the x and y axes, so that $\sum_{q \in Q} q_x = \sum_{q \in Q} q_y = 0$. Under this assumption, all the entries in the α_{RO} (sixth) row and column vanish except the lower right entry. Rearranging the rows and columns in the order r_x, α_{PA}, r_y, α_{TI}, α_{RO} (which does not affect the determinant or the condition number) reveals that the remaining matrix is block diagonal. Moving the sum inside the matrix and normalizing yields

$$\begin{bmatrix} 1 & a & 0 & 0 & 0 \\ a & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & a & 0 \\ 0 & 0 & a & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where}$$

$$a = \frac{\mu}{\sqrt{\mu^2 + \sigma^2}}, \quad \mu = \frac{1}{N} \sum_{q \in Q} q_x, \quad \sigma^2 = \frac{1}{N} \sum_{q \in Q} (q_x - \mu)^2. \tag{3-20}$$

Here N is the number of points q in Q, μ is the mean of the q_z values, and σ is the standard deviation. The determinant and condition number of this matrix are easily calculated; they

are

$$\det([\overline{\Gamma}_{ij}]) = (1 - a^2)^2 = \left(\frac{\sigma^2/\mu^2}{1 + \sigma^2/\mu^2}\right)^2, \quad \text{and}$$

$$\kappa_2([\overline{\Gamma}_{ij}]) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|} = \frac{1 + a}{1 - a} = \frac{(1 + a)^2}{1 - a^2}; \quad \text{from which}$$

$$\frac{1}{\sqrt{\det([\overline{\Gamma}_{ij}])}} \leq \kappa_2([\overline{\Gamma}_{ij}]) \leq \frac{4}{\sqrt{\det([\overline{\Gamma}_{ij}])}}.$$

Thus, if the standard deviation σ of the q_z values is small relative to their mean μ, the determinant will be small and the matrix will be ill-conditioned, and vice-versa. In the limiting case, if the z values are constant, then the matrix is singular.

Under perspective, equation (3-5) holds, so referring to equations (3-12) and (3-13),

$$D_g\mathbf{v} = (D_g\mathbf{u})(D_u\mathbf{v}) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ -q_z & 0 & q_x \\ 0 & -q_z & q_y \\ -q_y & q_x & 0 \end{bmatrix} \begin{bmatrix} f/u_z & 0 \\ 0 & f/u_z \\ -u_x f/u_z^2 & -u_y f/u_z^2 \end{bmatrix}.$$

Multiplying and taking advantage of the fact that $\mathbf{u} = \mathbf{q}$ because α_{PA}, α_{TI}, and α_{RO} are all zero, we obtain

$$D_g\mathbf{v} = f \begin{bmatrix} -1/q_z & 0 \\ 0 & -1/q_z \\ q_x/q_z^2 & q_y/q_z^2 \\ -(q_x^2 + q_z^2)/q_z^2 & -q_x q_y/q_z^2 \\ -q_x q_y/q_z^2 & -(q_y^2 + q_z^2)/q_z^2 \\ -q_y/q_z & q_x/q_z \end{bmatrix}.$$

Multiplying this matrix by its transpose yields the unnormalized geometric covariance matrix:

$$\frac{[\Gamma_{ij}]}{f^2} = \sum_{q\in Q} \begin{bmatrix} \frac{1}{q_z^2} & 0 & \frac{-q_x}{q_z^3} & \frac{q_x^2+q_z^2}{q_z^3} & \frac{q_x q_y}{q_z^3} & \frac{q_y}{q_z^2} \\[2mm] & \frac{1}{q_z^2} & -\frac{q_y}{q_z^3} & \frac{q_x q_y}{q_z^3} & \frac{q_y^2+q_z^2}{q_z^3} & -\frac{q_x}{q_z^2} \\[2mm] & & \frac{q_x^2+q_y^2}{q_z^4} & \frac{-q_x(q_x^2+q_y^2+q_z^2)}{q_z^4} & \frac{-q_y(q_x^2+q_y^2+q_z^2)}{q_z^4} & 0 \\[2mm] & & & \frac{(q_x^2+q_z^2)^2+q_x^2q_y^2}{q_z^4} & \frac{q_x q_y(q_x^2+q_y^2+2q_z^2)}{q_z^4} & \frac{q_y}{q_x} \\[2mm] & & & & \frac{(q_y^2+q_z^2)^2+q_x^2q_y^2}{q_z^4} & \frac{-q_x}{q_x} \\[2mm] & & & & & \frac{q_x^2+q_y^2}{q_z^2} \end{bmatrix}$$

Again making the simplifying assumption that the points q in Q are symmetrically placed about the x and y axes, rearranging the entries in the order r_x, α_{PA}, r_y, α_{TI}, r_z, α_{RO}, taking the sum inside the matrix, and normalizing, we obtain the perspective analog of equation (3-20):

$$\begin{bmatrix} 1 & b_x & 0 & 0 & 0 & 0 \\ b_x & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & b_y & 0 & 0 \\ 0 & 0 & b_y & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{where}$$

$$b_x = \frac{\sum_{q\in Q} c_x d}{\sqrt{\sum_{q\in Q} c_x^2 + q_x^2 q_y^2/q_z^4}\sqrt{\sum_{q\in Q} d^2}} \le \frac{\sum_{q\in Q} c_x d}{\sqrt{\sum_{q\in Q} c_x^2}\sqrt{\sum_{q\in Q} d^2}}$$

$$b_y = \frac{\sum_{q\in Q} c_y d}{\sqrt{\sum_{q\in Q} c_y^2 + q_x^2 q_y^2/q_z^4}\sqrt{\sum_{q\in Q} d^2}} \le \frac{\sum_{q\in Q} c_y d}{\sqrt{\sum_{q\in Q} c_y^2}\sqrt{\sum_{q\in Q} d^2}} \tag{3-21}$$

$$c_x = \frac{q_x^2+q_z^2}{q_z^2}, \quad c_y = \frac{q_y^2+q_z^2}{q_z^2}, \quad \text{and } d = \frac{1}{q_z}.$$

For brevity the following discussion refers only to b_x; symmetric statements apply to b_y. The right hand sides of the inequality in (3-21) represents a measure of the linear correlation
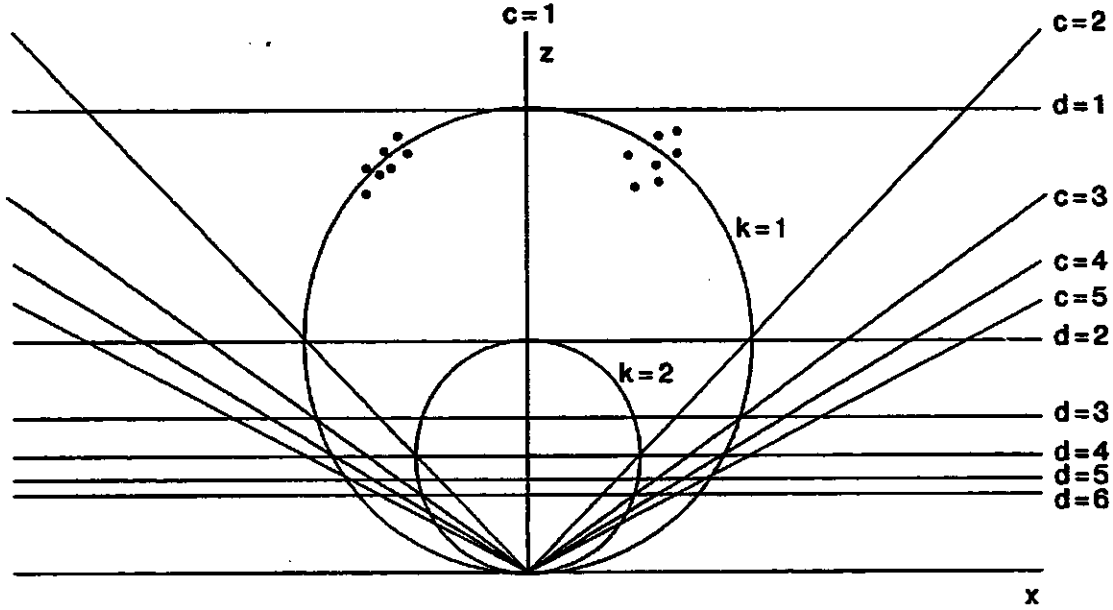
Figure 3-4.   Circles are loci of points $c_x = kd$ for various k, where $c_x = (x^2 + z^2)/z^2$ and $d = 1/z$. Horizontal lines show various values of d, diagonal lines various values of c_x. Cluster of three-space points on the left are ill-conditioned, because they mostly lie on such a locus; points on the right are better conditioned. This is a top view, that is x vs. z; a similar statement applies to y vs. z.

between c_x and d. This is a non-centralized correlation measure: it depends on the means μ_{c_x} and μ_d as well as on the standard deviations σ_{c_x} and σ_d. We can make the following statements about b_x: If $c_x = kd$ for some constant k, then $b_x = 1$ and the matrix is singular. At the other extreme, if c_x and d are independent, then

$$b_x = \frac{\mu_{c_x}}{\sqrt{\mu_{c_x}^2 + \sigma_{c_x}^2}} \frac{\mu_d}{\sqrt{\mu_d^2 + \sigma_d^2}}$$

Note that this is just a product of quantities like a in (3-20), except with respect to $c_x = (q_x^2 + q_z^2)/q_z^2$ and $d = 1/q_z$ instead of with respect to q_z.

In general, b_x will be large if most points q lie on a solution curve of $c_x = kd$ for some k, and small if they are spread out. These curves are just circles containing the origin, as shown in 3-4. Since large b_x implies that the matrix is nearly singular, and small b_x implies that it is well conditioned, the same conclusion holds in the case of perspective as in orthography: the matrix will be well-conditioned if the object is relatively three-dimensional compared to the distance from the camera to the object.

3.8. Solving for the parameters independently

Some problems require solving for subsets of the parameters while holding the remaining parameters constant, or for solving for the parameters as if they were independent of each other. For example, in all of the experiments the photometric parameters β and γ were not actually used; that is, they were held constant while some subset of the other parameters was varied. As another example, in some applications, for example a robot rover, all six geometric degrees of freedom many not be present; a rover might be constrained to planar motion, which is characterized by three of the six parameters. To solve for a subset of the parameters, one need only eliminate those rows and columns corresponding to the parameters which are to be held constant, and invert only the smaller matrix remaining. This can also be thought of as replacing the block of the matrix corresponding to the parameters to be held constant with the identity matrix.

Another useful alteration allows one to solve for parameters independently of each other. This is done by setting all the off-diagonal elements to zero. This is equivalent to solving for each parameter as if it were the only one that were to be changed, but in fact changing all the parameters at once. Solving for the variables independently has the advantage that eliminates the problem of the conditioning of the matrix, as the resulting matrix has a condition number of 1. This can be done in a more general way. Suppose for example that we wanted to solve for r_x, r_y, and r_z as a group, and independently but simultaneously to solve for α_{PA}, α_{TI}, and α_{RO} as a group. This is done by zeroing all elements whose row is from one group and whose column is from the other group. This allows us to decouple the equations but still retain some degree of relationship between the parameters. This might be desirable because the r_x, r_y, r_z matrix is usually well-conditioned, as is the α_{PA}, α_{TI}, α_{RO} matrix. Whether this technique produces good results is an open question.

3.9. Obtaining the reference image and z values

The remaining problems are obtaining a reference image I_1, the reference points p in the reference image, and their z values $z(p)$. In a real application, this will be performed as a training step, so it is not essential that it be fully automated. The technique presented here is to use a stereo pair consisting of the reference image and an additional image, the *training image*. The training image serves to provide depth information for the reference points, and is discared after the training step.

One approach is to manually select matching points and then use a stereo program such as that developed by Gennery (1980) to solve for their distances. The points for matching may be also be selected by an interest operator, such as that of Moravec (1980), and then matched against points in the other image by hand. The next chapter gives more details of the implementation of such a scheme.

Another approach obtains matching lines, which provide many matching points. This is done as follows: the operator locates a linear feature in the reference image, and identifies

its endpoints to the training program. The operator then locates the corresponding linear feature in the other image, and likewise identifies its endpoints. Now, on the assumption that the linear feature on the image is in fact linear in three-space, we have a whole line of reference points p, and from the known baseline one can easily calculate $z(p)$ for each of these points. Moreover, the points thus selected are especially desirable reference points, since points near intensity edges provide the most information for our matching technique. These z values can be fine-tuned to correct for operator inaccuracy and to provide sub-pixel positioning. The details of this operation are discussed in the next chapter as well.

3.10. Summary

This chapter has developed in detail the theory necessary to apply the method of differences to the optical navigation problem. This section summarizes the contributions of the chapter.

First the applications of optical navigation were surveyed to get an idea of the needs of the problem, and to show that the method as developed here was adequate to meet those needs. Then, starting with a precise definition of the camera model, equations were developed for estimating the parameters of that model. These equations were based on the method of differences as developed in the previous chapter. The calculations necessary for the implementation of those equations were discussed; the primary conclusion is that calculating the matrix requires by far the most computation, and is where special-purpose hardware could best be put to use. The conditions under which the inversion of the matrix is numerically well-conditioned were developed. This involved developing an estimate of the conditioning of the matrix based on geometric considerations, independent of the photometry of the scene. From this it was determined that the navigation problem is better conditioned when the objects in view are more three-dimensional. This geometric criterion for the numerical stability of the problem is applicable not only to the method of differences but to any match-based navigation technique.

The next chapter presents the results of experiments demonstrating the feasibility of using this technique for optical navigation. These experiments verify the theoretical analysis of the conditioning of the problem, and answer questions about the range of convergence and accuracy of the method.

Chapter 4

Optical Navigation:
Experiments

4.1. Introduction

This chapter presents experimental results obtained from applying the technique for optical navigation described in the previous chapter. The goal is to assess the accuracy of the method, its numerical stability, and the range of convergence that can be expected. These questions are of vital concern in both the rover and the visual servoing applications discussed in the previous chapter.

4.2. The data

Our data consist of views of two scenes: a computer-generated house scene and a real scene provided by Moravec from the Stanford cart. In the following discussion, we shall refer to the stereo pair consisting of the reference image and the training image as the *training pair*, and the stereo pair consisting of the reference image and the test image as the *test pair*. In every case, the left image of a pair is the reference image, and for these images the reference image in the training pair serves also as the reference image in the test pair.

Applying the method of differences to optical navigation requires that a set of reference points p and their distances $z(p)$ be determined. In the experiments described in this chapter, the points and their distances were determined by hand. In some real applications, such a manufacturing scenario, this procedure would be perfectly acceptable, since it would be part of a training step performed once. In other applications, such as a roving vehicle, an automated procedure would be necessary. The method of differences itself would of course be one candidate for this. However, no automated procedure was tried in order understand the behavior of the navigation algorithm in isolation without the confusing factor of imperfect reference points and distances.

Synthetic data. These experiments use two stereo pairs of the house scene: one from far away with a lens of long focal length, the *House 1* pair, Figure 4-7; and one from

nearby with a lens of short focal length, the *House 2* pair, Figure 4-8. These scenes are intended to explore the convergence properties of the algorithm free from the uncertainties in geometry and photometry of a real situation. The convergence range in the one-, two-, and multi-parameter cases is considered. In addition, the effect of adding noise to simulate photometric deviation from ideal is explored. In both cases, the same image served as both the training and the test image (namely, the right image of each pair); thus accuracy was essentially perfect, and so the house data were used only to assess the range of convergence and the stability of the algorithm.

The set P of points p used as reference points in the reference image were chosen in a semi-automated way as follows, using a program designed for the task. The goal was to choose reference points p near edges that would provide the most information for the method. An operator (the author), using cursors on a display screen, indicated the endpoints of three straight lines: the two forming the borders between the house and the roof and the one forming the border between the two visible walls. The operator then assigned z values to these endpoints by moving a cursor along the corresponding epipolar line in the other image to the match point. The reference points p consisted of the centers of all pixels near these lines (within $\frac{1}{2}$ pixel of each); there were approximately 250 such points. Then the program assigned to each reference point p a distance $z(p)$ by linearly interpolating the z values assigned by the operator to the endpoints of the lines. Then for each reference point p, its z value $z(p)$ was independently fine-tuned using the method of differences in the simple one-dimensional form of equation (2-2). (See Section 5.2 for a detailed explanation of this adjustment procedure). This ensured that each reference point was matched against a point of equal intensity somewhere along its epipolar line. This procedure was designed to mimic a procedure that could be used to select reference points in a application (such as a manufacturing situation) where the reference image is fixed.

Real data.   To assess the accuracy of the method in the presence of imperfect data (as well as the questions of stability and convergence) the cart data were used. The cart experiments are based on three real images of the same scene: a left, a middle, and a right image. By considering the left against the middle image, we obtain a short-baseline stereo pair, the *Cart S* pair, Figure 4-9; by considering the left against the right image we obtain a long-baseline stereo pair, the *Cart L* pair, Figure 4-10. These two stereo pairs are used as the training pairs to determine the distances $z(p)$ of a set of reference points p by the procedure described below. Then, in the long-baseline *Cart L* case the middle picture serves as the test image, and in the short-baseline *Cart S* case the right picture serves that role; in both cases, the left picture of course serves as the reference. Thus, unlike in the synthetic image case, the test image is different from training image. This of course introduces error but is more realistic, especially for assessing the accuracy.

The cart images were taken by cameras that suffered from certain geometric distortions; polynomials for correcting these distortions were calculated by Moravec, and the cart images were resampled to eliminate the distortion. However, the coefficients of the correcting

polynomials could be regarded as additional geometric parameters to be solved for by the method of differences, along with the position and orientation parameters; no attempt to do this has been made.

The reference points p and distances $z(p)$ were determined as follows. Visually identifiable reference points p were selected in the left (reference) picture by two methods: by hand (20 points) and by the Moravec interest operator (50 points); in addition, the set consisting of the union of these two (70 points) was used. The Moravec operator automatically picked desirable edge points; the hand-picked points were also near edges. The matching points v (see Figure 3-1 on page 47) in the middle and right pictures were selected by hand. Matching points for all of the hand-selected points p were found, but many of the points selected by the interest operator had to be discarded because of occlusion (including occlusion by the edge of the viewport) or ambiguity. The six sets of matching points (hand, interest, or union points; and short- or long-baseline pair) were given to a camera model solver written by Gennery (1980). (The function of Gennery's program should not be confused with the program discussed here: Gennery's program is not a matching program, but rather takes matches as input.) The camera model solver rejected an additional point from some of the sets, and produced six sets of camera models and distances $z(p)$. The distances $z(p)$ could be determined because the distances between the cameras that took the pictures was accurately known. In addition, these sets of distances $z(p)$ were adjusted by the same procedure described for the synthetic data; this procedure resulted in the rejection of some more points, because of low gradient or great photometric discrepancy. The final result was twelve sets of reference points p and distances $z(p)$, ranging from 9 points to 35 points. This allows us to assess the effect on accuracy of size of baseline, number of points, and whether the z values were adjusted. For the stability and convergence investigations, only the *Cart L* pair with adjusted z values were used.

The remainder of this chapter is divided into four sections. Section 4.3 presents some experimental results verifying our prediction of the conditions under which the matrix to be inverted in solving for the camera parameters is numerically stable. Section 4.4 considers solving for the camera parameters in three cases of increasing complexity: the one-parameter case, the two-parameter case, and the multi-parameter case. Section 4.5 investigates the accuracy of the method using the cart data. Finally, Section 4.6 summarizes the conclusions reached in these experiments.

4.3. Matrix conditioning

The previous chapter discussed the conditions under which the inversion of the matrix $[\bar{\Sigma}_{ij}]$ given by equation (3-10) on page 49 is numerically stable. There were two main predictions: first, that the geometric conditioning of the problem is a good indicator of the actual (photometric) conditioning, and second (based on the first) that the matrix is better conditioned for nearby and for relatively three-dimensional objects. These predictions are verified by the results presented in Table 4-1. This table shows the condition numbers of

House 1			House 2			Cart L			
Geo	5 × 5	17 × 17	Geo	5 × 5	17 × 17	Geo	5 × 5	17 × 17	
1288	822	757	34	34	21	14	10	12	α_{PA}, r_z
1741	1357	1442	50	56	65	13	11	10	α_{TI}, r_y
1835	5157	3919	50	74	75	14	33	18	α_{PA}, α_{TI}, r_z, r_y
2910	5695	5527	55	148	159	23	56	22	α_{PA}, α_{TI}, α_{RO}, r_z, r_y
3018	6416	6497	66	188	204	27	52	23	α_{PA}, α_{TI}, α_{RO}, r_z, r_y, r_z
6	32	17	5	30	14	2	4	2	r_z, r_y, r_z
5	4	3	2	3	4	4	3	3	α_{PA}, α_{TI}, α_{RO}
1482	839	761	45	66	32	15	11	12	α_{PA}, r_z, r_z

Table 4-1. Table shows condition number of matrix. First part is for the *House 1* scene, second part for the *House 2* scene, and third part for the *Cart L* scene. Columns are for condition number of geometric matrix $[\Gamma_{ij}]$, and for photometric matrix $[\overline{\Sigma}_{ij}]$ for picture with two sizes of smoothing windows. Rows are for various combinations of parameters solved for.

---

the geometric and photometric matrices for the *House 1* and *House 2* scenes. That the geometric conditioning is closely related to the condition number of the actual photometric matrix is shown by the similarity between the three columns for each scene. The effect of the spatial configuration is shown by the relatively large condition numbers for the *House 1* scene, the smaller numbers for the closer view of the same object in the *House 2* scene, and the still smaller numbers for the *Cart L* scene, in which several objects at different distances were present. Note also that the presence of highly correlated variables such as x distance r_z and pan α_{PA} in the set of variables solved for leads to larger condition numbers. This is of course in agreement with the theoretical prediction.

The condition number represents the most that the relative error in the matrix will be multiplied by to obtain the relative error in the calculated solution (see equation (3-15) on page 53). Thus how large a condition number is acceptable depends on two factors: how accurately the matrix is calculated, and how large an error in the calculated solution is acceptable.

The probable accuracy of the calculation of the matrix was assessed by looking at the effect of adding (relatively large amounts of) noise to the pictures on the accuracy of the matrix, for the *House 2* scene. In particular, referring to (3-15), we have $A = [\overline{\Sigma}_{ij}]$, and thus $\Delta A = \Delta[\overline{\Sigma}_{ij}]$ is the error introduced into the matrix by adding noise to the images. The relative error introduced into the matrix is given by $\|\Delta[\overline{\Sigma}_{ij}]\|/\|[\overline{\Sigma}_{ij}]\|$. Table 4-2 shows the results. These values should not be taken too literally as they are samples of random quantities, since the noise added is random. The implications of these values are discussed below.

20 db S/N		10 db S/N		0 db S/N		
5×5	17×17	5×5	17×17	5×5	17×17	
0.00068	0.0011	0.0025	0.0038	0.0037	0.0059	$\alpha_{PA},\ \mathbf{r}_x$
0.00051	0.00082	0.0025	0.0036	0.00002	0.011	
0.00014	0.00015	0.00042	0.00044	0.00036	0.00005	$\alpha_{TI},\ \mathbf{r}_y$
0.00014	0.00009	0.0012	0.00021	0.00078	0.00004	
0.0083	0.0022	0.015	0.0068	0.076	0.027	$\alpha_{PA},\ \alpha_{TI},\ \mathbf{r}_x,\ \mathbf{r}_y$
0.014	0.0029	0.0089	0.0026	0.17	0.11	
0.0082	0.0020	0.027	0.0063	0.13	0.037	$\alpha_{PA},\ \alpha_{TI},\ \alpha_{RO},\ \mathbf{r}_x,\ \mathbf{r}_y$
0.018	0.0077	0.053	0.014	0.18	0.071	
0.0080	0.0042	0.031	0.012	0.12	0.034	$\alpha_{PA},\ \alpha_{TI},\ \alpha_{RO},\ \mathbf{r}_x,\ \mathbf{r}_y,\ \mathbf{r}_z$
0.0088	0.015	0.037	0.020	0.21	0.046	
0.0038	0.0042	0.024	0.011	0.14	0.036	$\mathbf{r}_x,\ \mathbf{r}_y,\ \mathbf{r}_z$
0.0019	0.0047	0.026	0.013	0.18	0.11	
0.0098	0.0020	0.031	0.0061	0.18	0.037	$\alpha_{PA},\ \alpha_{TI},\ \alpha_{RO}$
0.0055	0.0047	0.038	0.015	0.27	0.060	
0.0013	0.0032	0.010	0.0095	0.078	0.026	$\alpha_{PA},\ \mathbf{r}_x,\ \mathbf{r}_z$
0.0023	0.0034	0.018	0.012	0.12	0.010	

Table 4-2. Table shows effect of noise on photometric matrix $[\overline{\Sigma}_{ij}]$. Values given in the table are relative error, $\|\Delta[\overline{\Sigma}_{ij}]\|/\|[\overline{\Sigma}_{ij}]\|$, for various signal-to-noise ratios (S/N), smoothing windows, and combinations of parameters used in the matrix. The noise levels are, left to right, moderate (standard deviation of signal (house image) is 10 times noise), large (standard deviation of signal is $\sqrt{10}$ times noise), and extremely large (standard deviation of signal is equal to standard deviation of noise). For each entry, two values are given: the top one represents the same uniformly distributed random noise field being added to the picture in each case, to allow a fair comparison between the cases; the bottom one represents a different noise field in each case, to get an idea of the variability of the relative error, since the quantities reported are in fact samples of random quantities. Because they are samples of random quantities, these values should not be taken too seriously.

As for the second factor, a large relative error in the calculated parameter adjustments is tolerable because they will be used in an iterative scheme. For example, suppose (in the one-parameter case) that the correct parameter value is p, and that the current parameter value is $\hat{p} = p + e$. Then the value we wish to calculate is $x = -e$ (where x is like x in (3-15)). If the computed $x + \Delta x$ (where Δx is the error as in (3-15)) was anywhere between 0 and $-2e$ then the new estimated camera parameter would be between $\hat{p} - e$ and $\hat{p} + e$, which would be an improvement assuming symmetry. This range is guaranteed if the

relative error $|\Delta x/(x + \Delta x)|$ is no larger than $\frac{1}{2}$.

Thus, even with moderate amounts of noise, which the table shows may result in a relative error in the calculated matrix of on the order of 0.01, a condition number of on the order of 50 is tolerable. For *Cart L* scene this is essentially attained in all cases; for the *House 2* scene it is attained in the two-, and three-parameter cases investigated, marginally so for the four-parameter case, but not for the five- and six-parameter cases; for the *House 1* scene it is attained only for the position-only and orientation-only cases. This would seem to suggest that the five- and six-parameter cases are not stable for the *House 2* scene, but the experiments discussed below suggest that even extremely large amounts of noise are not likely to impair its stability even in the six-parameter case. It must be remembered that the estimates given are extremely pessimistic worst-case estimates, representing the tail of the probability distribution of possible relative error; thus they must be taken with a grain of salt. More informative estimates would make statements about the probabilities of various errors, but the analysis seems intractable. Experience must serve as the final judge.

It should be noted, referring to Table 4-1, that the geometric conditioning is a reasonable estimate of the stability of the matrix, generally not being off by more than a factor of two to three. This means that the geometry of a situation can be used to assess its numerical stability independent of the particular images encountered later in actual use.

4.4. Range of convergence

This section explores the range of convergence provided by the algorithm in three cases of increasing complexity: the one-parameter case, the two-parameter case, and the many-parameter case.

One-parameter case.   In the one-parameter case, all parameters but one are given the correct value and held constant; thus the performance of the algorithm for values near the correct value of the parameter of interest is explored. The "matrices" in this case are 1×1 matrices, and so numerical stability is not an issue. Figure 4-3 shows how the one-parameter results are presented. This figure is similar to Figure 2-2 on page 36.

Figures 4-11 through 4-16 present the results for the *House 1* scene; Figures 4-17 through 4-22 present the results for the *House 2* scene; Figures 4-23 through 4-26 present the results for the *House 2* scene with two levels of noise (for α_{PA} and α_{TI} only); and Figures 4-27 through 4-32 present the results for the *Cart L* scene.

The region of convergence R for each of the parameters and each of the scenes is presented as an interval in Table 4-4. This region was calculated using the ideas discussed in Section 2.6 starting on page 29. A value of 0.9 was chosen for the c of equation (2-49) and Figure 2-2: this generally allows convergence within about 10 iterations, and making c larger does not in general appreciably enlarge the region R, because the error estimate generally falls off to zero fairly quickly. This is the value illustrated in Figures 2-2 and 4-3.
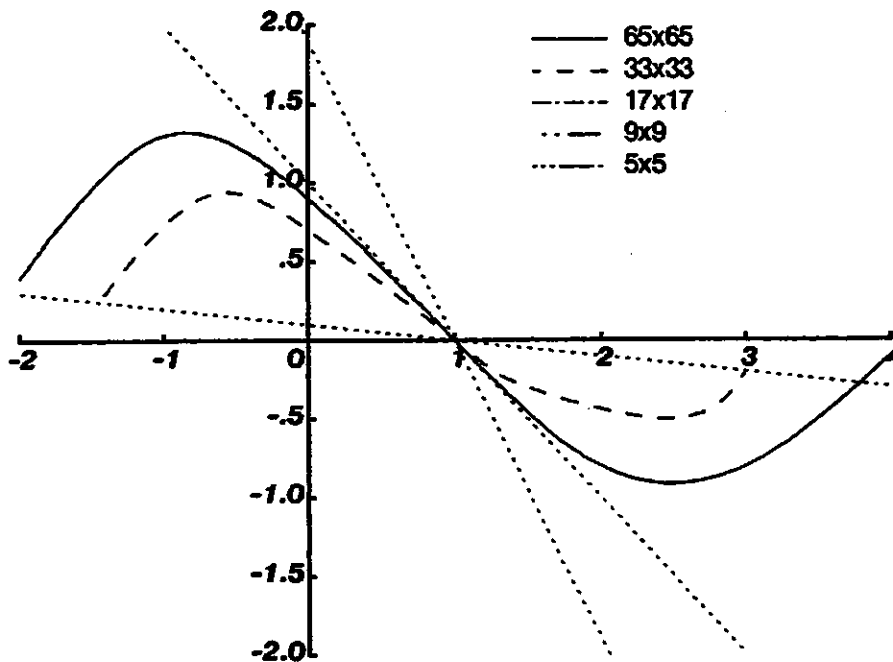
Figure 4-3. One-parameter results are displayed in graphs like this. In each case, a single parameter is solved for. The initial value of this parameter is shown along the horizontal axis; the vertical axis shows the computed delta for the parameter. The ideal case is a line of slope −1 (the middle dotted line) crossing the horizontal axis at the correct value of the parameter (1.0 in this example). A typical actual case is shown by the solid curve above; near the actual value of the parameter it is almost a straight line, but as the initial value gets further from the correct value, the computed delta falls back toward zero. In each case several curves are shown for various smoothing windows applied to the image (only two curves are shown above, but all five line types listed in the key apply to all the graphs of actual data). The line of slope −1 is not shown in the following figures, but in each case the slopes of the various curves near where they cross the horizontal axis is approximately −1. The other two dotted lines have slopes of −0.1 and −1.9; their significance is explained in the text.

These results illustrate several trends. The most obvious is that a larger smoothing window results in a larger range of convergence. This is in accord with both intuitive expectations and theoretical calculations. Second, a wider angle lens results in a wider range of convergence for the angular parameters, for a given size smoothing window; this is illustrated by comparing the *House 1* and *House 2* scenes, which differ only in the focal length of the (simulated) lens used. This is of course because with a wider angle of view,

House 1		House 2		Cart L		
33 × 33	65 × 65	33 × 33	65 × 65	33 × 33	65 × 65	
− 10 +4.0	− 15*+ 14	− 20 + 34	− 50*+ 42	− 6 + 11	− 13 + 12	α_{PA}
−9.2 +2.2	− 15*+ 11	− 22 + 10	− 42 + 46	− 6 + 10	− 12 + 19	α_{TI}
− 19 + 40	− 35 + 40*	− 30*+ 29	− 30*+ 30*	− 22 + 30*	− 28 + 30*	α_{RO}
−4.2 +1.2	−5.9 +3.0*	−0.8 +2.7	−3.7 +3.2	−125*+110	−125*+ 90	r_x
−2.9 +0.7	−3.0*+2.5*	−1.1*+0.5	−0.9 +2.5	−110 + 60	−120 +125*	r_y
− 10*+6.8	− 10*+ 10*	− 4*+1.3	−3.6 +1.7	−380 +270	−420 +260	r_z

Table 4-4. Table shows convergence region as an interval around the convergence value. Angles are in degrees; others are in arbitrary units for the house scenes, centimeters for the cart scene.

---

a given angular disparity results in a smaller pixel disparity in the images; since it is the pixel disparity that determines whether convergence will be possible, this means that a wider angular disparity is tolerable. Finally, comparing the natural *Cart L* scene against the artificial *House 1* scene, we see that the large angle of view of the *Cart L* scene has not resulted in any larger region of convergence for the angular parameters. This is because the natural scene is more cluttered and has more detail; these details make the high-order terms of the Fourier series larger, and as our previous analysis predicts, this reduces the range of convergence. Presumably, this can be overcome by increasing the size of the smoothing window.

Two-parameter case. Let us now turn to experiments in which matrices are actually solved. This section considers the two-parameter case before proceeding to the multi-parameter case in the next section, because the two-parameter case is easier to present graphically.

Graphs like Figure 4-3 represent the change in parameter estimate calculated by the method of differences. A similar presentation in the two-parameter case would entail a four-dimensional graph. Nevertheless, the two-parameter data can be presented in a useful form on a two-dimensional graph. Imagine a grid of equally spaced points; each point, used as an initial estimate for two parameters, is carried to a new point by the method of differences. Ideally, each point would be carried to the point representing the correct values of the parameters, and so the original grid would be squeezed down to a single point. In reality, each point is carried to some position that is generally closer to the correct point than its initial position, and so the original grid gets distorted to a new irregular grid. Since this grid is usually too irregular to make any sense to the eye, we in fact graph the grid that results from moving each point halfway from its starting position to its position as updated by the method of differences. Thus, in the ideal case, the initial grid is transformed into a

grid half the initial size, i.e. halfway between the initial grid and a single point. In addition, we can examine the transformation of the grid after some number of iterations, to get an idea of the convergence behavior.

Three examples of such grids are shown in Figures 4-33, 4-34, and 4-35. The pluses in these figures represent the corners of the starting grid; the solid curves are the grids after one iteration; and the dotted curves are the grids after five iterations. Because many of the points have nearly converged after five iterations, the dotted grids roughly outline the correct answer in which every point is carried to the correct value. In Figure 4-33, the fourth point from the left in the bottom row shows over-correction on the first iteration: its position inside the ideal grid roughly outlined by the dotted lines reveals the over-correction. Note that the bottom right-hand point has still not approached its correct position after 5 iterations, and thus seems unlikely to converge. Figure 4-34 shows that the *House 2* scene displays a larger convergence range, as one would expect due to the wider angle of the lens. The top row of points however appear not to converge. Finally, Figure 4-35 shows a range of convergence that is somewhat elliptical in shape. This is because of the confusion that exists between the two parameters being investigated. For example, a large pan to the right can be tolerated if it is "compensated" for by an x motion to the left; of course the x motion can't completely nullify the effect of the pan (otherwise they would be indistinguishable and the matrix would be singular). The net result of the combined motions is a relatively small motion on the test image of the points corresponding to the reference points, and it is the total error in position on the test image plane that determines whether convergence is possible.

This method of presentation shows the trends exhibited by the method of differences, but it doesn't clearly show over what region the algorithm will converge. To this end, we introduce diagrams like Figure 4-36. This diagram shows for each point on a grid whether that point converged to the right value (square), to the wrong value (circle), or didn't converge at all (plus). The convergence test was based on the size of the adjustment at each step; if the adjustment ever fell below a certain threshold, the point was deemed to have converged. In each diagram the large box shows the convergence range predicted if each parameter were operating independently as in Table 4-4.

Figure 4-36 shows convergence for the *House 2* pair for the largest smoothing window used; 4-37 shows the equivalent information for the *Cart L* pair. Note that while not every point in the box converges to the correct value, there are in addition some points outside the box that do converge to the correct value. The box based on the independent parameters serves as a rough prediction of the two-parameter case.

It should be emphasized that it is not necessary that the algorithm converge at every possible point representing a possible initial guess. In particular, the method of differences can be coupled with a search; all that is required is that one of the initial guesses lie inside the region of convergence. Thus the grid on which the search is done must be guaranteed to have a point inside the region of convergence.

Figures 4-38 and 4-39 illustrate the relationship between the various convergence values. In Figure 4-38, the large symbols are represent points that converge to the correct value; in Figure 4-39, the large symbols are those converging to one particular incorrect value, namely the one represented by the dot. This shows that the plane (i.e. the two-parameter space) is divided up essentially contiguous regions of points all converging to the same value. Between these regions, representing local maxima or false "peaks", are "rivers" where no convergence occurs.

Finally, Figure 4-40 further illustrates the elongated shape of the region of convergence for two similar parameters, α_{PA} vs. r_x in this case. This point was also mentioned previously. The line running roughly through the middle of the region might be called a "locus of confusion" for the parameters, and the correlation coefficient of the set of points represents some sort of measure of the confusibility of the two parameters.

Multi-parameter case.   Now we turn to the more difficult problem of examining the range of convergence for the multi-parameter case. Let us assume that the regions of convergence in a multi-parameter space are multi-dimensional blobs akin to the two-dimensional regions in the two-dimensional parameter spaces examined previously. The approach is to look at two-dimensional slices of these blobs. This is accomplished by examining the convergence given initial values of the parameters lying on a grid in a two-dimensional parameter space, as before, but in addition the algorithm is allowed to adjust the value of one or more additional parameters.

For example, compare Figure 4-37, in which α_{PA} and α_{TI} are investigated, with Figure 4-41; in the latter figure, three parameters α_{PA}, α_{TI}, and r_x are adjusted by the algorithm; but the initial values of this three-dimensional parameter space that are investigated (i.e. used as input to the algorithm) are only those lying in the same grid in α_{PA}-α_{TI} space as those in 4-37. Thus, Figure 4-41 shows a two-dimensional slice of a three-dimensional blob of convergences. Note that this cross-section has a shape roughly similar to the shape in the two-parameters case, but it is smaller. The six-parameter case is shown in Figure 4-42. The region of convergence is smaller still, but of similar shape.

Finally, another approach is illustrated in Tables 4-43 and 4-44. In these tables we examine each corner of a six-dimensional hypercube in camera-parameter space. The second of these tables uses a hypercube that is 50 percent larger than the first. The tables show that in the first case convergence is attained at nearly all values, while in the second case some of the corners of the hypercube lie outside of the region of convergence.

The primary conclusion to draw from all of this is that while convergence is not attained at all of the points that would be implied by considerations of one-dimensional convergence alone, a substantial portion of them do converge.

4.5. Accuracy

The accuracy of the method was assessed by giving the parameters the known correct value and allowing the method of differences to adjust one or more of them. The difference between the initial correct value and the final value was taken to be the error. The correct values were obtained as follows. Since the cart pictures were taken by a single camera moving on an accurate slider, the distance between the origins of the coordinate systems for the images is accurately known: it is 26 cm between the left and middle images, and 52 cm between the left and right images. The other five parameters (azimuth, elevation, pan, tilt, and roll) are not as accurately known; the values calculated by Gennery's program from the matching points are used. For example, when the long-baseline training pair *Cart L* is used, the five parameters obtained using the right-hand image as the training image together with a range of 26 cm (expressed as α_{PA}, α_{TI}, α_{RO}, r_x, r_y, and r_z) are used as the correct values for the position of the middle test image, to be compared with the values obtained by the program; and vice-versa for the short-baseline training pair *Cart S*.

In addition, for comparison with the house scenes, the accuracy is tested using the same image as both the training and test images: the right image is used as the test image on points obtained using the *Cart L* pair, and the middle image on points using the *Cart S* pair. Let us refer to this as the *homogeneous* case, because the same image serves as both the test and training image; the house experiments were all homogeneous. By contrast, let us refer to the case in which the training and test images are different as the *heterogeneous* case. Both the procedure described in the preceding paragraph and most real applications are heterogeneous cases.

Thus six variables are to be investigated: long- vs. short-baseline training pair; homogeneous vs. heterogeneous application; adjusted vs. non-adjusted distance values; the source of the reference points: hand-picked, interest operator-picked, or the union of these two sets; the size of the smoothing window; and the number of parameters to be solved for. As for the last variable, two sets of parameters are used: one of all six parameters (the robot arm case), and of three variables (the rover case: α_{PA}, r_x, and r_z). In addition, the various sources of points and procedures for using them serve to produce a spectrum of sizes of the reference point set, making for a total of seven factors investigated.

The results of these experiments, one of the cases above per line, for a 9×9 smoothing window are displayed in Table 4-5. In each case, all six parameters were given the correct value as their initial value; then the three or six under investigation were adjusted by the method of differences for 20 iterations, or until convergence was reached (as determined by the size of the adjustment), whichever came first. One conclusion can be drawn immediately from this data: for all practical purposes the accuracy in the angles was perfect, in all cases. Further conclusions are more easily drawn by presenting part of this same data in a graphical form, as in Figure 4-6. This figure shows the error in the x position for the various cases, and thus represents only one column of Table 4-5.

Several conclusions can be drawn from this figure. First, the homogeneous case (indicated by dotted lines) is generally better than the heterogeneous case. This fact is neither

α_{PA}	α_{TI}	α_{RO}	r_x	r_y	r_z	
-0.01	0.00	0.00	3.29	0.00	-4.40	h-r
0.00	0.00	0.00	3.12	0.00	1.63	i-r
0.00	0.00	0.00	1.69	0.00	0.78	u-r
0.02	0.00	0.00	-3.72	0.00	-17.31	h-a
-0.02	0.00	0.00	2.13	0.00	6.78	i-a
0.00	0.00	0.00	1.30	0.00	6.55	u-a
0.00	0.00	0.00	1.74	0.00	-0.75	h-r θ
0.00	0.00	0.00	0.25	0.00	0.68	i-r θ
0.00	0.00	0.00	0.42	0.00	0.26	u-r θ
0.02	0.00	0.00	-5.82	0.00	-19.47	h-a θ
-0.07	0.00	0.00	-0.07	0.00	3.76	i-a θ
-0.01	0.00	0.00	2.45	0.00	-0.21	u-a θ
-0.01	0.00	0.00	3.67	0.00	-2.65	h-r
0.00	0.01	0.01	1.53	-2.52	4.94	i-r
0.00	0.00	0.00	1.48	-0.02	1.89	u-r
0.04	-0.01	-0.02	-14.57	5.29	-18.84	h-a
-0.01	0.01	0.00	1.06	-2.93	8.58	i-a
0.00	0.00	0.00	1.42	0.36	3.93	u-a
0.00	0.00	-0.01	-2.86	-1.37	-5.29	h-r θ
0.00	0.00	0.00	0.49	1.26	-2.62	i-r θ
0.00	0.00	0.00	0.81	-0.34	-1.54	u-r θ
0.02	-0.03	-0.10	4.36	23.59	-55.20	h-a θ
-0.02	0.00	-0.01	2.72	-2.70	6.48	i-a θ
-0.01	0.00	0.01	2.62	-0.59	0.37	u-a θ
-0.01	0.00	0.00	0.21	0.00	0.58	h-r
-0.01	0.00	0.00	1.30	0.00	-3.12	i-r
-0.01	0.00	0.00	0.64	0.00	-0.99	u-r
0.00	0.00	0.00	-1.03	0.00	-0.58	h-a
0.00	0.00	0.00	0.86	0.00	1.70	i-a
0.00	0.00	0.00	0.45	0.00	0.90	u-a
-0.01	0.00	0.00	1.48	0.00	1.70	h-r θ
-0.01	0.00	0.00	0.38	0.00	-2.89	i-r θ
-0.01	0.00	0.00	0.60	0.00	0.35	u-r θ
0.00	0.00	0.00	2.23	0.00	-3.28	h-a θ
0.00	0.00	0.00	0.88	0.00	0.96	i-a θ
0.00	0.00	0.00	1.16	0.00	0.53	u-a θ
0.00	0.00	-0.02	-4.08	-1.22	-1.28	h-r
0.00	0.01	0.00	-0.57	-2.46	1.79	i-r
-0.01	0.00	-0.01	-1.19	-1.14	1.65	u-r
0.03	0.00	-0.03	-13.26	0.14	10.62	h-a
0.00	0.00	0.00	-0.31	-1.54	3.30	i-a
0.01	0.00	-0.01	-1.60	-0.77	1.54	u-a
0.01	0.01	-0.03	-5.16	-5.00	2.45	h-r θ
0.00	-0.01	-0.01	0.75	3.48	-9.27	i-r θ
-0.01	0.00	-0.01	-0.03	0.78	-4.29	u-r θ
0.00	0.00	0.01	2.11	-1.32	-1.89	h-a θ
0.00	0.00	-0.01	1.19	-0.04	-0.10	i-a θ
0.00	0.00	0.00	1.45	0.00	-0.40	u-a θ

Table 4-5. Table shows the error between the value after 20 iterations and the correct value, for each parameter. A 9×9 smoothing window was used. Angles are in degrees, others in centimeters.

surprising nor particularly useful. It is not surprising because the camera parameters for the training pair are taken as given, and the z values are adjusted for a match while the camera parameters are held constant; so in the homogeneous case when the same training pair is used as a test pair, in which the camera parameters are adjusted to attain a match, no adjustment of the camera parameters should be necessary to find the best possible match. This is true even though there might be slight error in the hand-picked matches and in the camera parameters, because the adjustment of the z values compensates for these er-
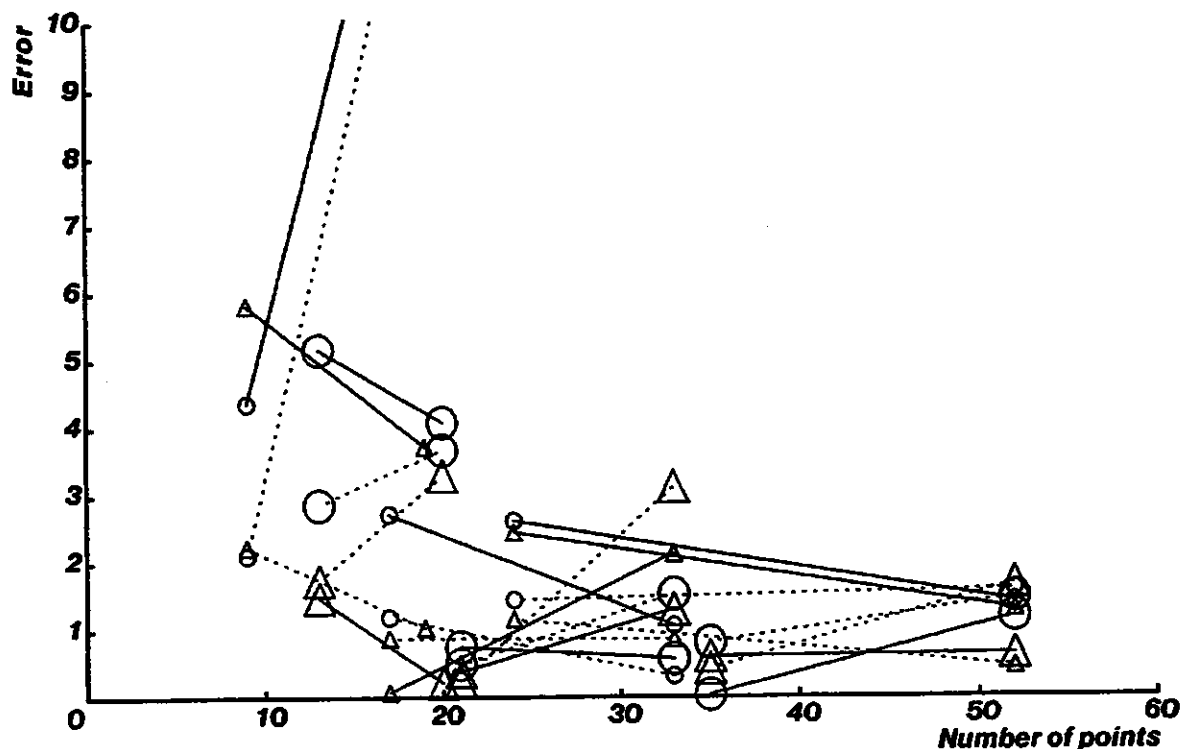
Figure 4-6. Graph shows the absolute error from the r_z column of Table 4-5. Each point represents the result with a set of reference points, distinguished by resulting error (in cm) on the vertical axis, and by number of reference points on the horizontal axis. Triangles indicate the case where three parameters were solved for, circles six. Large circles or triangles represent the long-baseline training pair, small symbols the short-baseline pair. At the right end of each line is a reference point set in which the adjustment of the z values by the method of differences was not carried out; the left end of each line represents the same set in which the adjustment was done (and thus points were pruned). Finally, dotted lines correspond to homogeneous experiments in which the test pair was the same as the training pair; solid lines are the heterogeneous case, in which the test pair was not the same as the training pair.

rors. That some adjustment is necessary is due to the fact that the camera parameters were in fact not perfect and a perfect match for each reference point was not attainable along its epipolar line during the adjustment procedure. This observation is not particularly useful because in a real application the homogeneous case is never encountered. Thus we concentrate on the data for the heterogeneous cases from now on.

Because the adjustment procedure also prunes points, the effects of adjustment and the effects of the number of points are difficult to separate. But a general trend can be observed, by looking at the slopes of the lines connecting the corresponding adjusted and non-adjusted cases: the adjustment procedure increases the accuracy unless it reduces the number of points by too much, in which cases it decreases the accuracy. This is consistent with the intuition that adjusting the z values during training should improve the accuracy, but that on the other hand better accuracy is obtained with more points.

The question of the effect of the number of parameters solved for on the accuracy is more difficult. There seems to be no consistent pattern, with sometimes three (indicated by triangle in Figure 4-6) parameters being more accurate and sometimes six (indicated by circle). However, the best accuracy with point set with the most points was obtained by the three parameter case. In any case, solving for six parameters simultaneously does not seem to be impractical, from the standpoint of accuracy.

The long vs. short baseline comparison is made by looking at the large vs. small symbols (connected to the solid lines, for the heterogeneous case) The long baseline seems to give somewhat more accurate results; in most (heterogeneous) cases, it provided the best accuracy.

It is important to realize that the error is relative not to length of the stereo baseline but to the distances to the reference points in three-space. That is, making the baseline longer while maintaining the same scene if anything should decrease the error in the parameter determination; on the other hand, scaling the whole scene up or down should scale the error with it. The point is that the error of a centimeter or so should not be compared with the baseline of 52 cm but rather with the distance to the reference points in the room-sized scene of several meters.

Refer to the table in Figure 4-10, which shows the relationship between a small change in each of the six parameters and the position on the test image of each matching point. Taking one example, the matching positions change at about 5 pixels per degree for α_{PA} and α_{TI}, $\frac{1}{2}$ pixel per centimeter for r_x and r_y, and $\frac{1}{7}$ pixel per centimeter for r_z. This means that a one pixel error in position on the screen means, on the average, a two-centimeter error in the x and y directions and a seven-centimeter error in the z direction. Thus, the error in x and y positions would be explained by an approximate one-pixel error in matching. Furthermore, this also explains the observed larger z error and very small angular error. Thus, these data support the following rule of thumb: the expected accuracy of parameter estimation is that obtained by assuming a one-pixel matching error.

4.6. Summary

This chapter has examined the conditioning of the method of differences as applied to optical navigation, the range of initial parameter value estimates over which the algorithm could be expected to converge to the correct parameter values, and the accuracy of the computed parameter values. This section summarizes the contributions of this chapter.

Table 4-1 shows that the geometric condition number was a good estimate of the actual (photometric) condition number. Theoretical investigation of the geometric condition number predicted that the matrices to be inverted are better conditioned when the three-space reference points are more "three-dimensional", for example when they are viewed from a closer, wider angle point of view than from a view that approximates orthography. This was verified by the comparison of the condition numbers associated with the three scenes in Table 4-1. Experiments were done in which substantial amounts of noise were added to the images. The conditioning of the matrices, shown in Table 4-1, was sufficiently good that the error in the matrices induced by the noise in the images, shown in Table 4-2, did not result in intolerably large error in the computed parameter values, especially when the method was used in an iterative scheme.

The range of convergence in the one-, two-, and multi-parameter cases was investigated. The results obtained in the one parameter case, shown in Table 4-4 were quite promising. They showed, for example, that method of differences could compute the actual position after a move of an autonomous rover even when the new position was in error by 10 to 50 of pan or tilt, depending on the scene, or 30 degrees of roll, independent of the scene. Position errors of 100 centimeters or more were tolerable. Solving for more than one parameter simultaneously reduced this range, but not excessively, as seen in Figures 4-36 and following.

These ranges depend on the degree of smoothing, as demonstrated in Table 4-4. Smoothing windows up to about $\frac{1}{4}$ the image size were used. Larger smoothing windows would increase the range of convergence up to a point, limited by the number of reference points still within the view of the other camera and by edge effects of smoothing. Increasing the angle of view of the camera would increase the allowable smoothing window and thus the range of convergence, but would decrease the accuracy if the number of pixels in the retina remained constant. This is because the parameter accuracy is ultimately limited by the approximate one pixel match accuracy provided by the method.

If a larger convergence range is needed, a search technique can be added; because of the relatively large region of convergence, the search can be fairly coarse. If the technique could be implemented in hardware to provide essentially continuous feedback, no search should be needed. This is especially promising for the robot arm scenario. The high speed of the algorithm on a VAX 11/780 suggests that real-time performance is obtainable on special-purpose hardware.

Finally, the accuracy, even on relatively poor data from the Stanford cart, was found to be quite good, as seen in Table 4-5 and Figure 4-6. This accuracy was attained even though the photometric parameters β and γ were not used. This is because the reference points were chosen to be near edges, where photometric error affects the match accuracy less. Positioning errors in the x and y directions of 1 or 2 centimeters in a room-sized scene were common, given about 50 reference points. Positioning errors in the z direction were larger. This suggests the possibility of a side-looking camera to obtain accuracy in the z direction. The angular parameters were estimated with extremely high accuracy. Experiments verify what one's intuition suspects: the more reference points, the more accurate the answer;

this is shown in Figure 4-6. Of secondary importance were such factors as the length of the baseline, whether adjustment of the z values of the reference points was done (by the method of differences), and the number of parameters solved for. The method seems to provide the parameter estimation accuracy expected with a one-pixel matching accuracy.
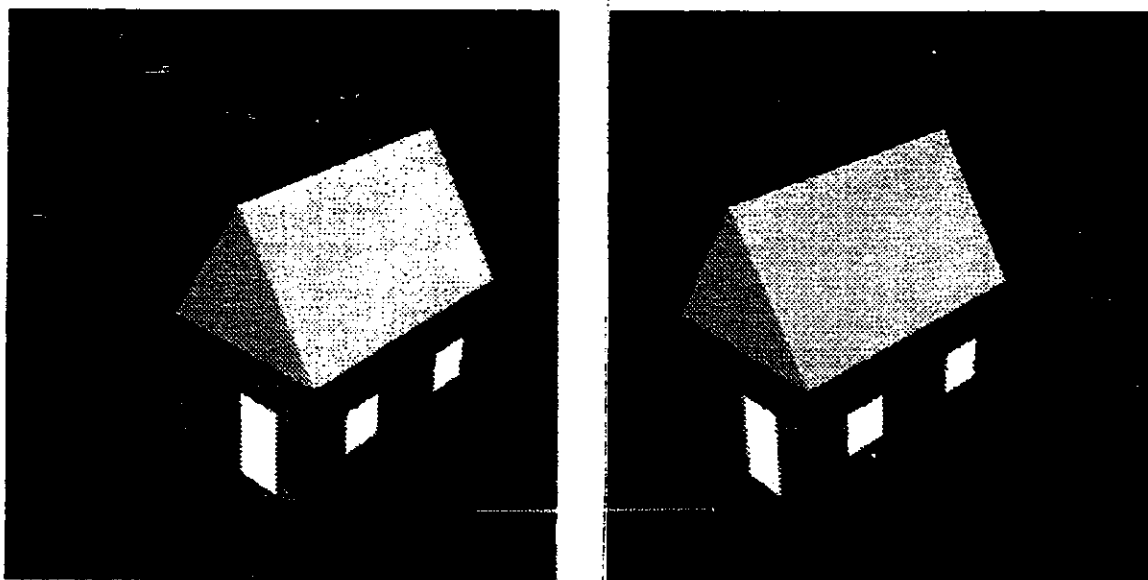
Picture size: 250 × 250 pixels (both images).
Focal length: 415 pixels (both images).

Parameter	Value	Min	Max	RMS
α_{PA}	0.0	7.24	7.60	7.33
α_{TI}	0.0	7.24	7.65	7.32
α_{RO}	0.0	0.34	1.80	1.09
r_x	2.0	20.09	24.85	22.99
r_y	0.0	20.09	24.85	22.99
r_z	0.0	1.06	5.65	3.46

Figure 4–7. *House 1.* Both images of stereo pair are shown.. Sizes of pictures and focal lengths are given. For each parameter, the value of that parameter is given; angles are in degrees, others in arbitrary units. The relationship between each parameter and pixels on the image is given, in pixels/degree for the angles, pixels/unit for the others. Since this relationship varies over the image (for example, for rotation and r_z translation it is zero at the center of rotation and focus of expansion respectively, larger further out), its minimum, maximum, and root mean square values are given.
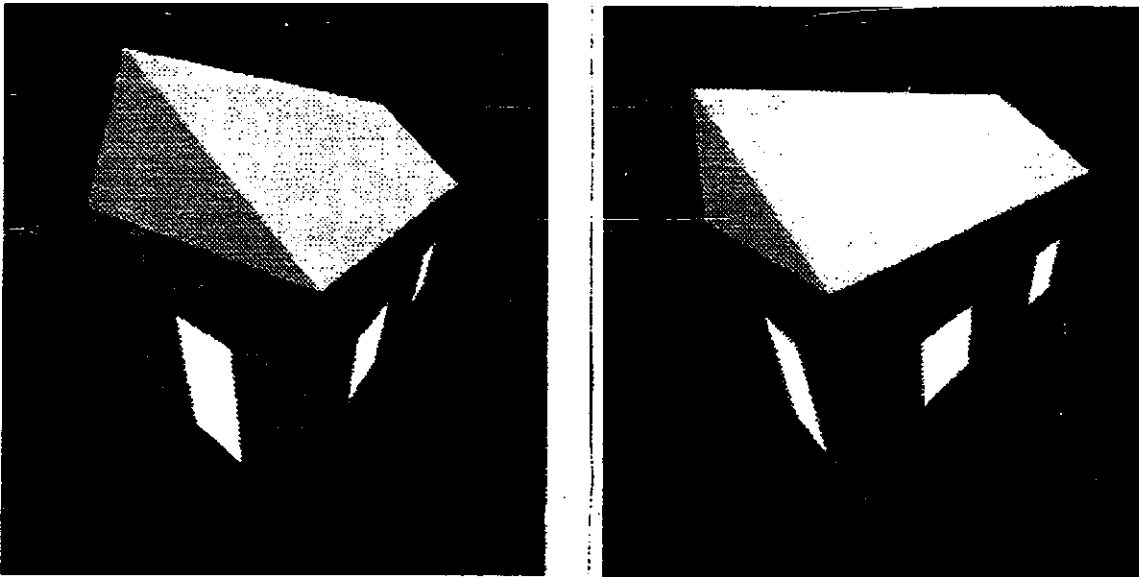
Picture size: 250 × 250 pixels (both images).
Focal length: 125 pixels (both images).

Parameter	Value	Min	Max	RMS
α_{PA}	−11.3	2.18	. 3.55	2.46
α_{TI}	0.0	2.18	3.52	2.39
α_{RO}	0.0	0.10	1.92	0.99
r_x	1.0	23.09	60.43	45.92
r_y	0.0	23.79	63.61	48.07
r_z	0.0	6.60	29.85	23.23

Figure 4-8. *House 2.* Both images of stereo pair are shown.. Sizes of pictures and focal lengths are given. For each parameter, the value of that parameter is given; angles are in degrees, others in arbitrary units. The relationship between each parameter and pixels on the image is given, in pixels/degree for the angles, pixels/unit for the others. Since this relationship varies over the image (for example, for rotation and r_z translation it is zero at the center of rotation and focus of expansion respectively, larger further out), its minimum, maximum, and root mean square values are given.

Picture size: 250 × 250 pixels (both images).
Focal length: 307 pixels (both images).

Parameter	Value	Min	Max	RMS
α_{PA}	−0.16	5.36	6.11	5.52
α_{TI}	−0.13	5.37	6.12	5.55
α_{RO}	−0.19	0.20	2.26	1.35
r_x	25.70	0.31	1.30	0.71
r_y	1.43	0.31	1.30	0.71
r_z	−3.66	0.03	0.50	0.18

Figure 4-9. *Cart S.* Both images of stereo pair are shown.. Sizes of pictures and focal lengths are given. For each parameter, the value of that parameter is given; angles are in degrees, others in arbitrary units. The relationship between each parameter and pixels on the image is given, in pixels/degree for the angles, pixels/cm for the others. Since this relationship varies over the image (for example, for rotation and r_z translation it is zero at the center of rotation and focus of expansion respectively, larger further out), its minimum, maximum, and root mean square values are given.

Picture size: 250 × 250 pixels (both images).
Focal length: 307 pixels (both images).

Parameter	Value	Min	Max	RMS
α_{PA}	0.76	5.36	6.27	5.58
α_{TI}	0.00	5.36	5.98	5.53
α_{RO}	0.07	0.28	2.79	1.42
r_x	51.92	0.23	1.14	0.57
r_y	1.24	0.23	1.14	0.57
r_z	−2.52	0.03	0.37	0.14

Figure 4-10. *Cart L.* Both images of stereo pair are shown.. Sizes of pictures and focal lengths are given. For each parameter. the value of that parameter is given; angles are in degrees, others in arbitrary units. The relationship between each parameter and pixels on the image is given, in pixels/degree for the angles, pixels/cm for the others. Since this relationship varies over the image (for example, for rotation and r_z translation it is zero at the center of rotation and focus of expansion respectively, larger further out), its minimum, maximum, and root mean square values are given.
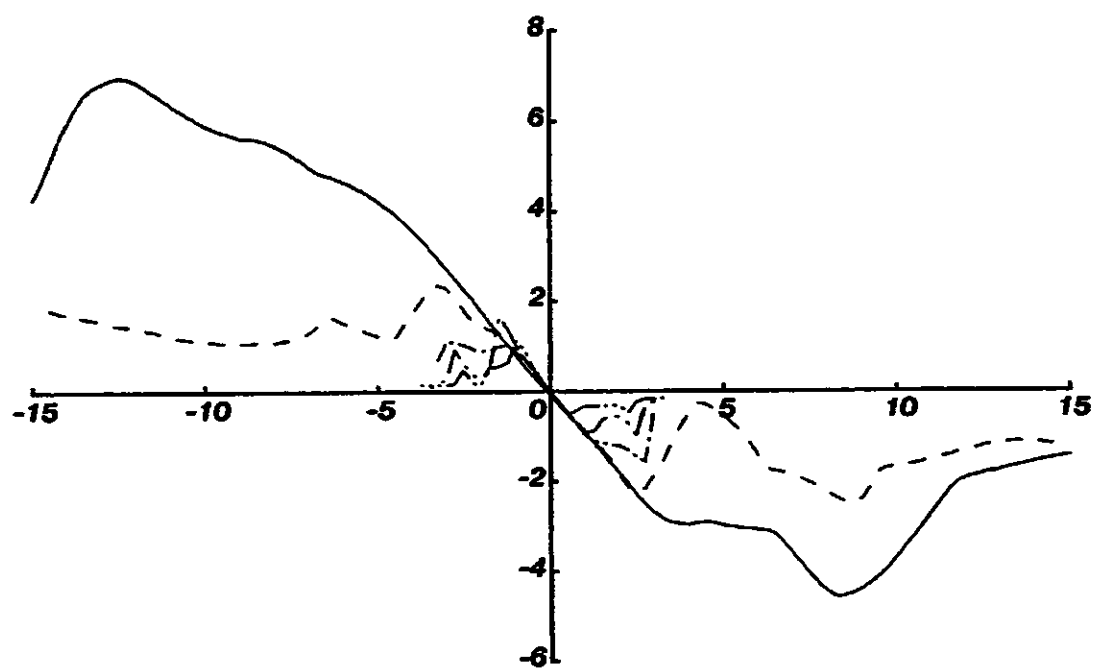
Figure 4-11. *House 1* pair, α_{PA}. See Figure 4-3 for interpretation.
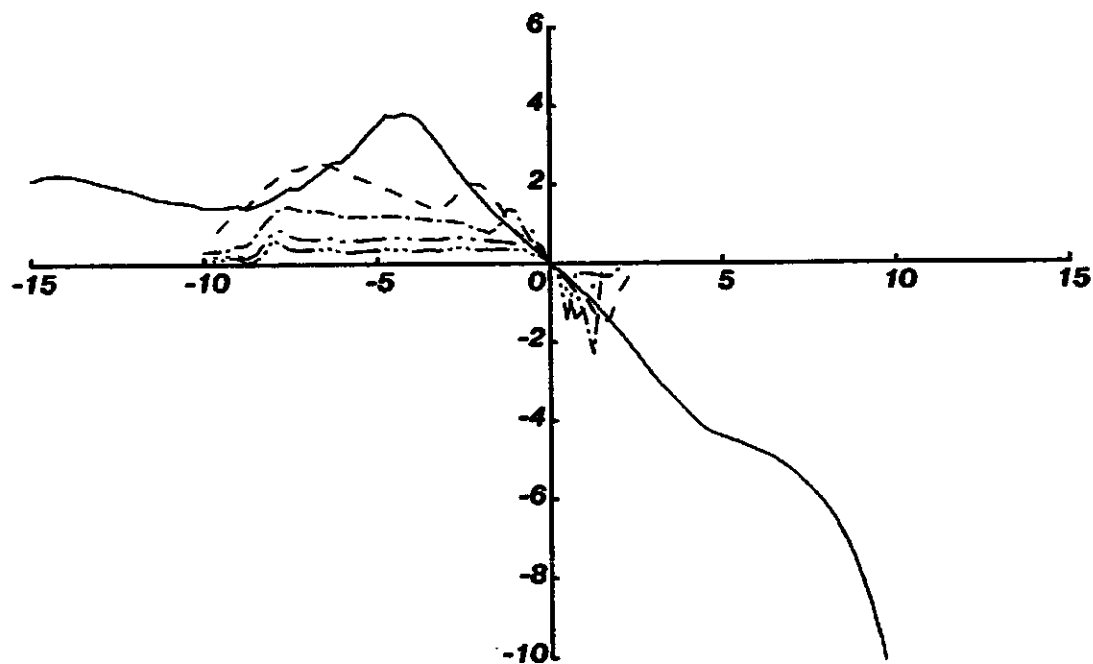
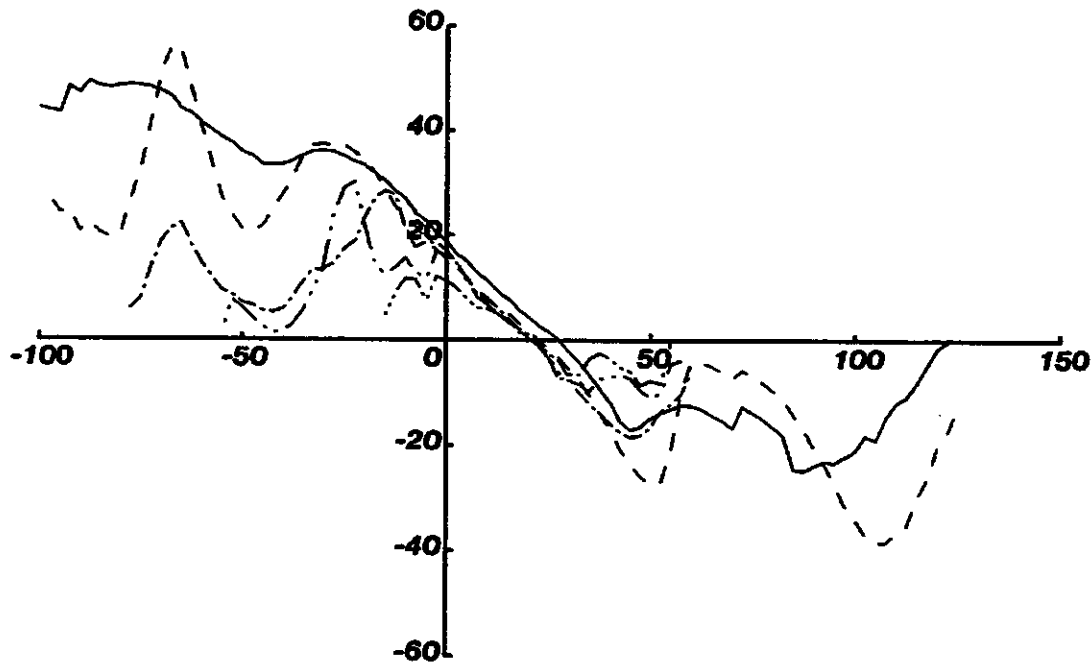Figure 4-12. *House 1* pair, α_{TI}. See Figure 4-3 for interpretation.
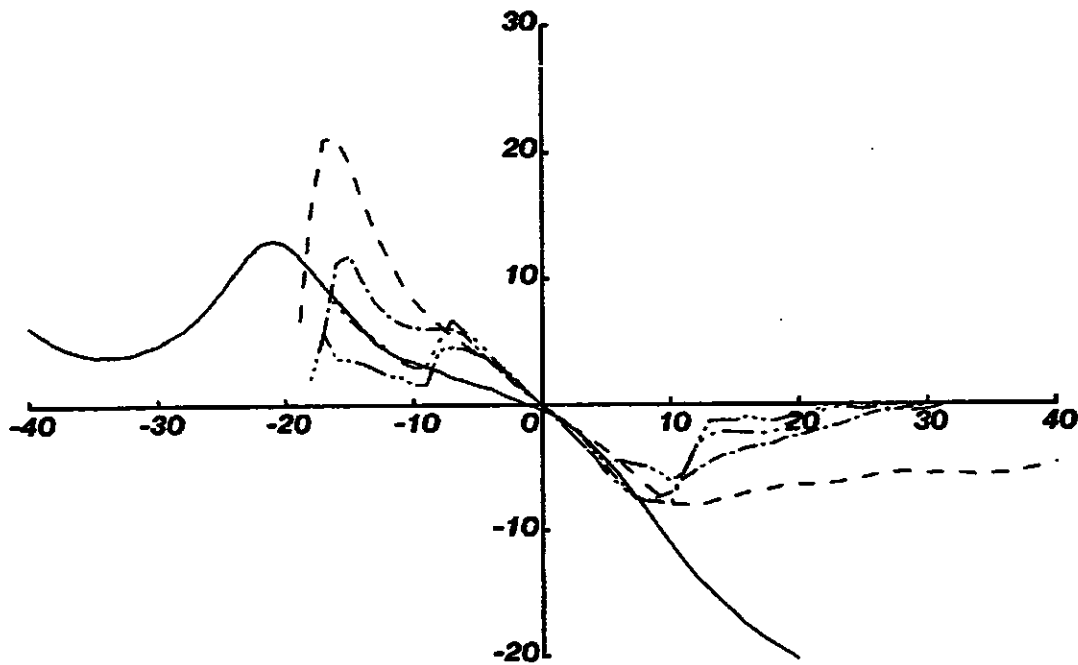
Figure 4-13. *House 1* pair, α_{RO}. See Figure 4-3 for interpretation.
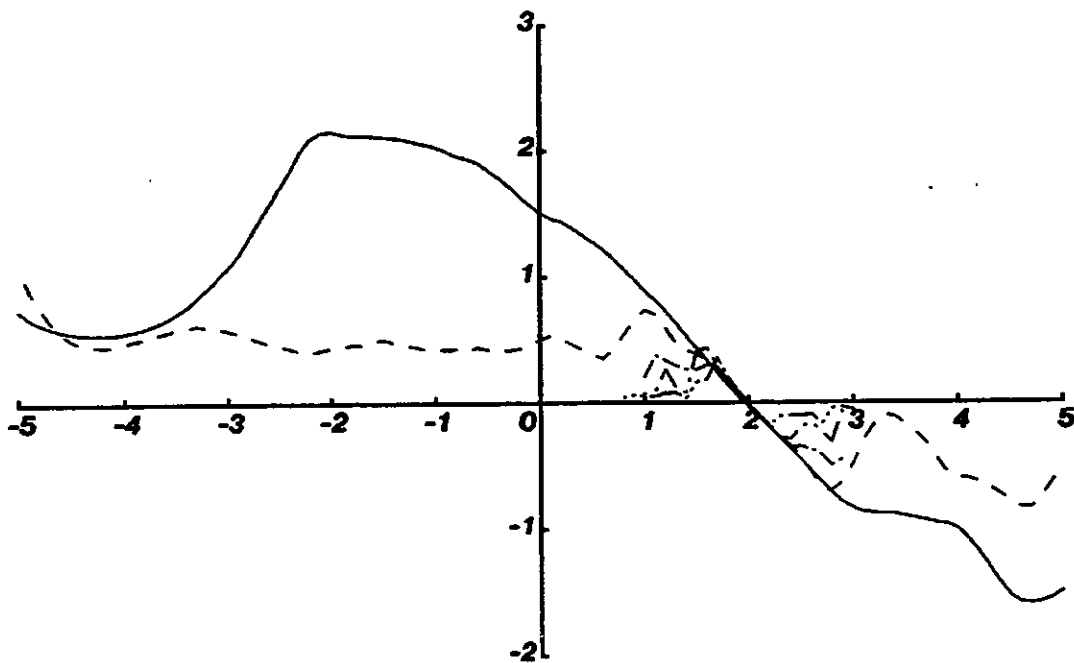
Figure 4-14. *House 1* pair, r_x. See Figure 4-3 for interpretation.

Figure 4-15. *House 1* pair, r_y. See Figure 4-3 for interpretation.
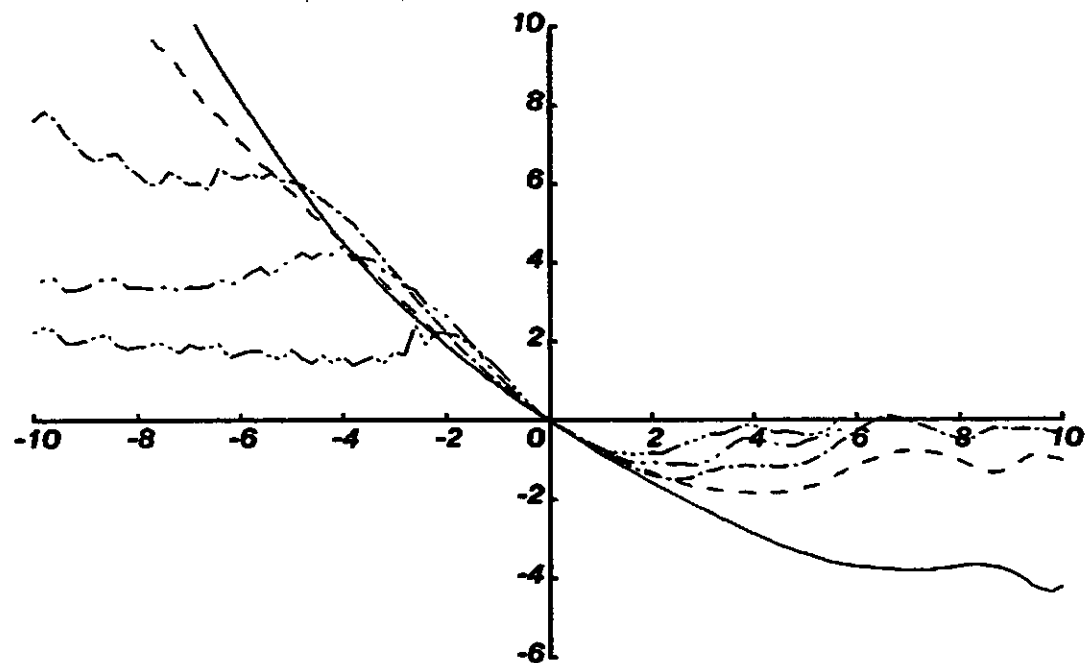
Figure 4-16. *House 1* pair, r_z. See Figure 4-3 for interpretation.
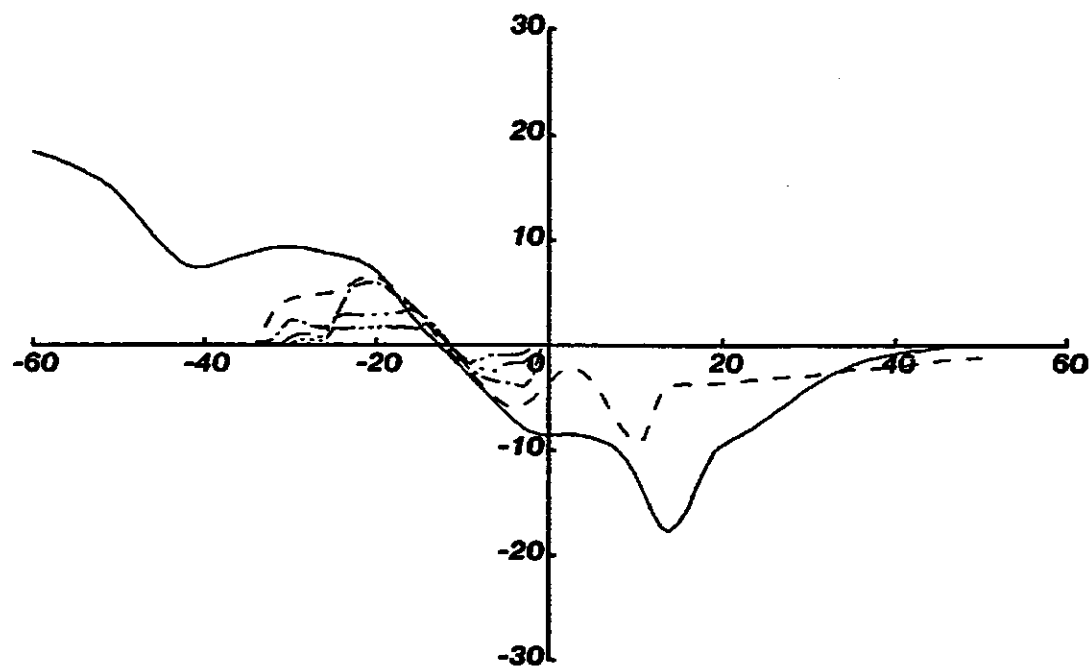
Figure 4–17. *House 2* pair, α_{PA}. See Figure 4–3 for interpretation.
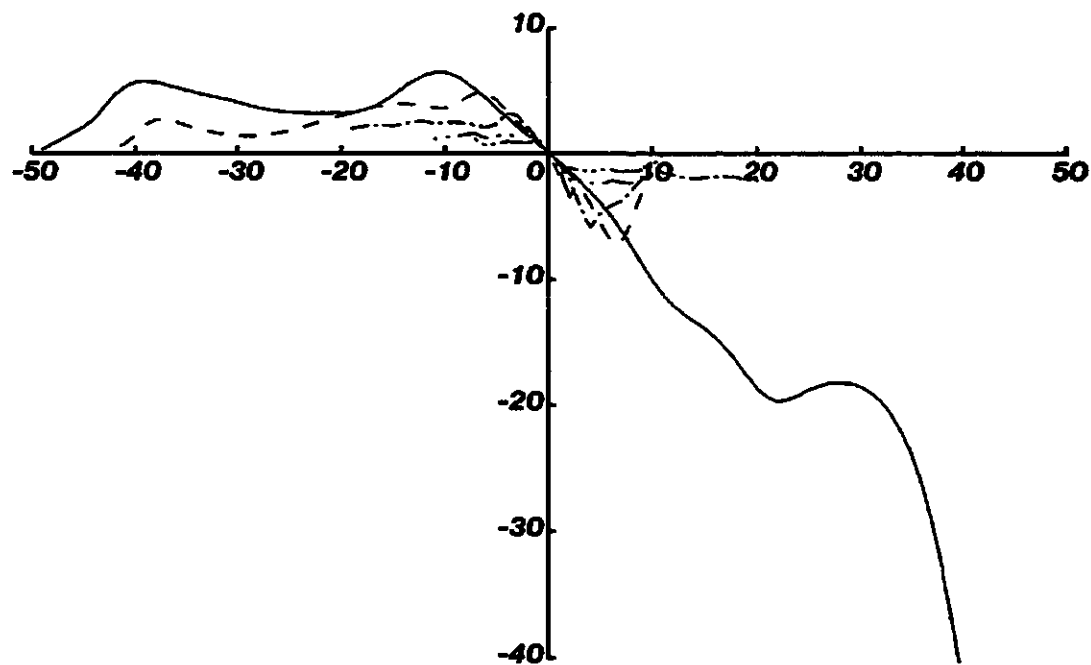
Figure 4–18. *House 2* pair, α_{TI}. See Figure 4–3 for interpretation.
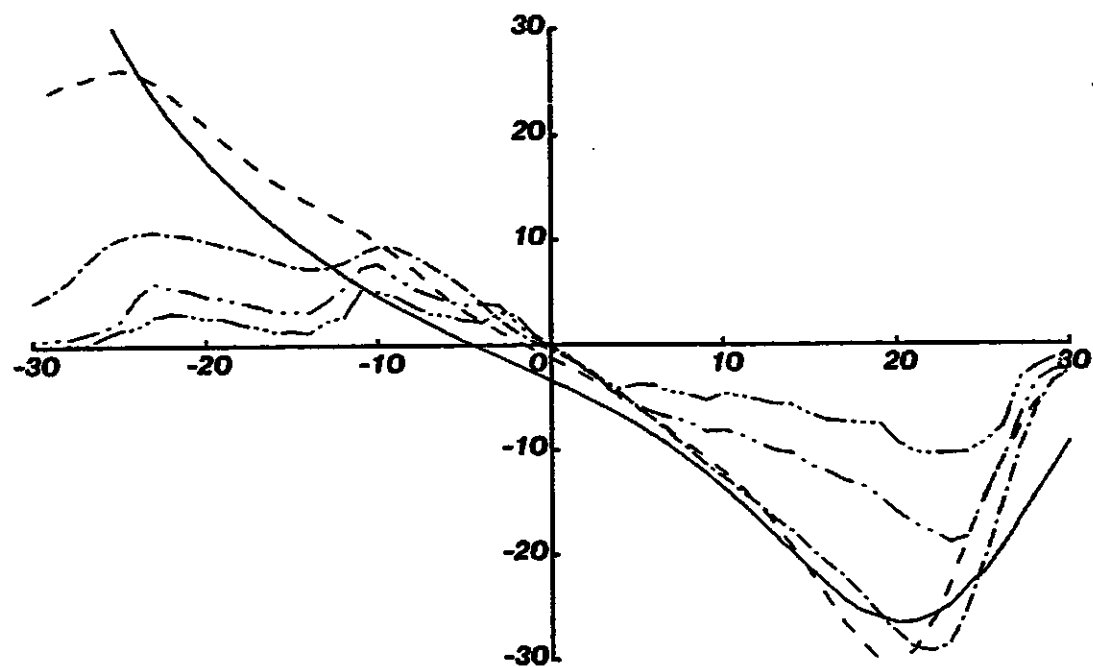
Figure 4-19. *House 2* pair, α_{RO}. See Figure 4-3 for interpretation.
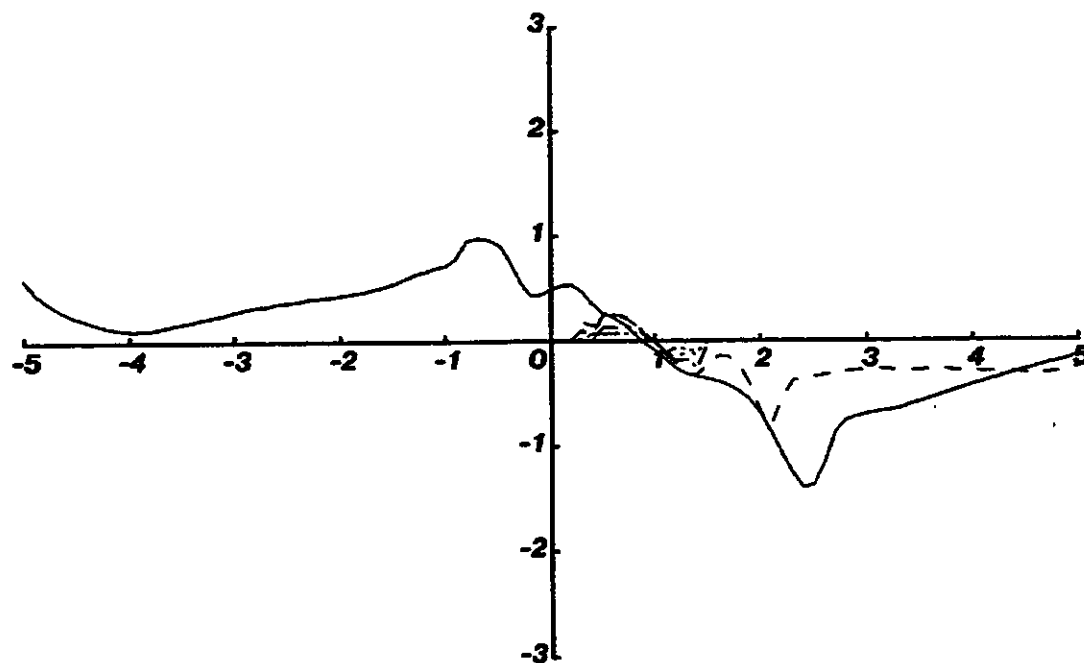
Figure 4-20. *House 2* pair, r_x. See Figure 4-3 for interpretation.
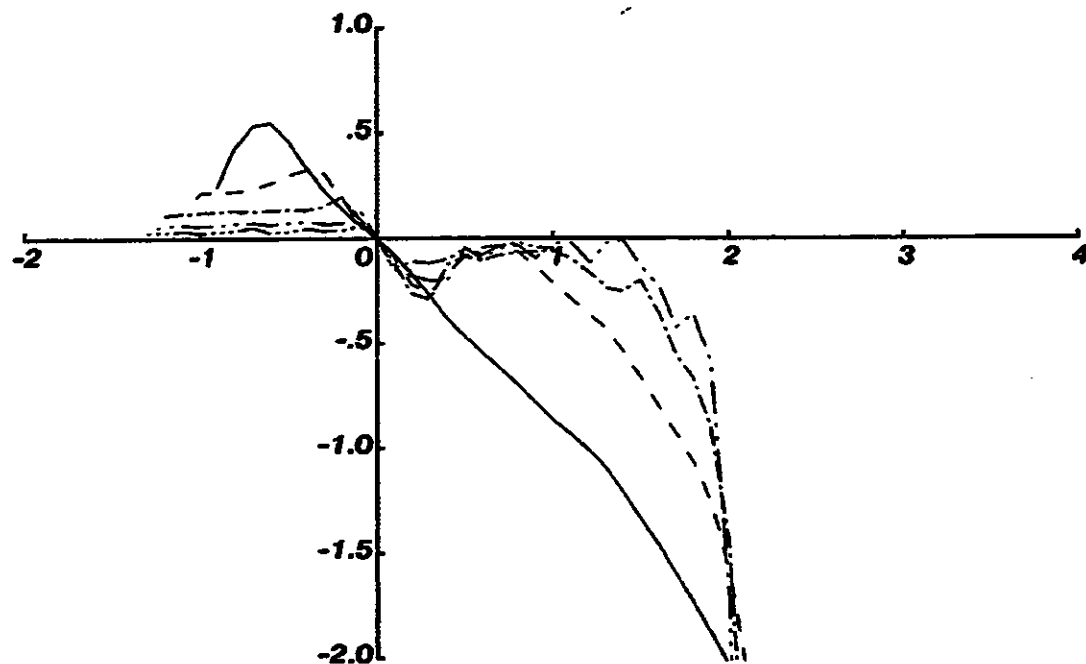
Figure 4–21. *House 2* pair, r_y. See Figure 4–3 for interpretation.
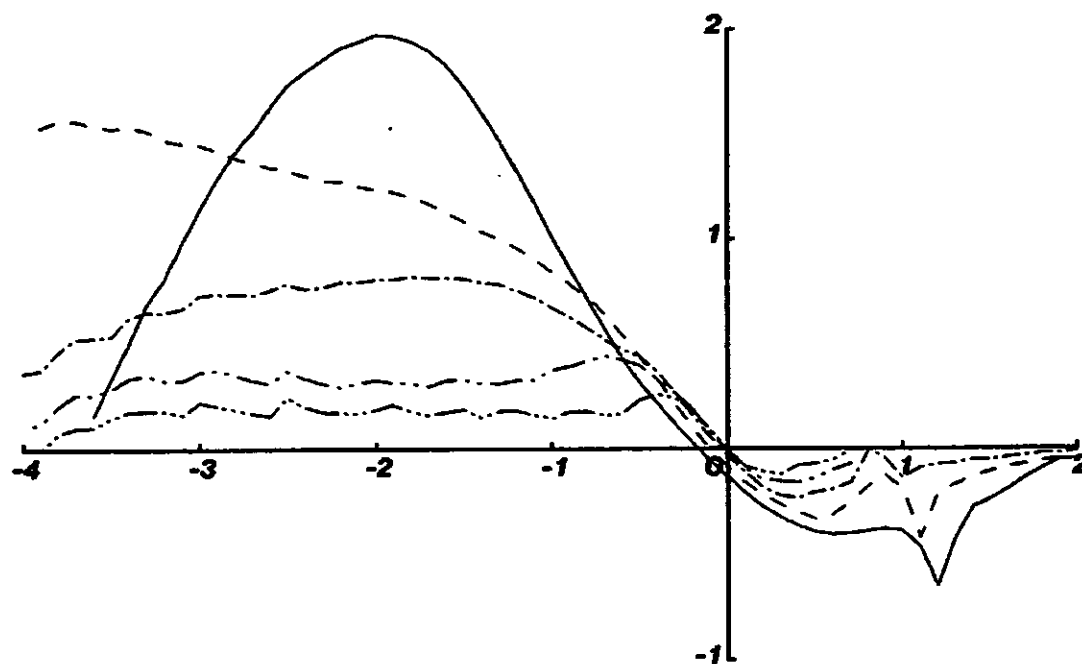
Figure 4–22. *House 2* pair, r_z. See Figure 4–3 for interpretation.
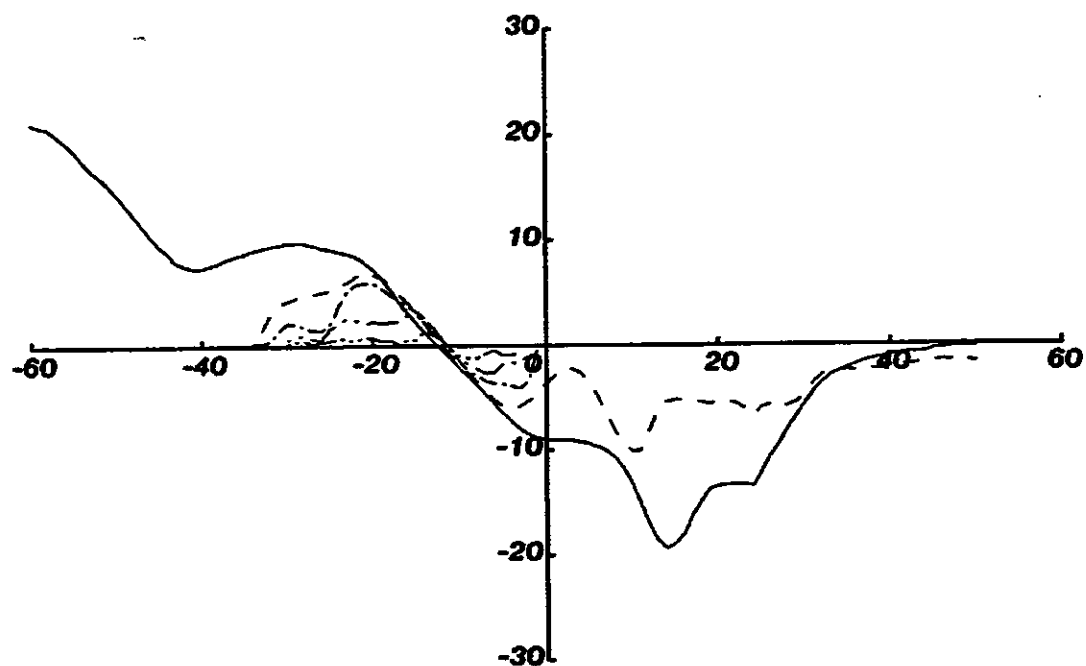
Figure 4-23. *House 2* pair, 10 *db* S/N, α_{PA}. See Figure 4-3.
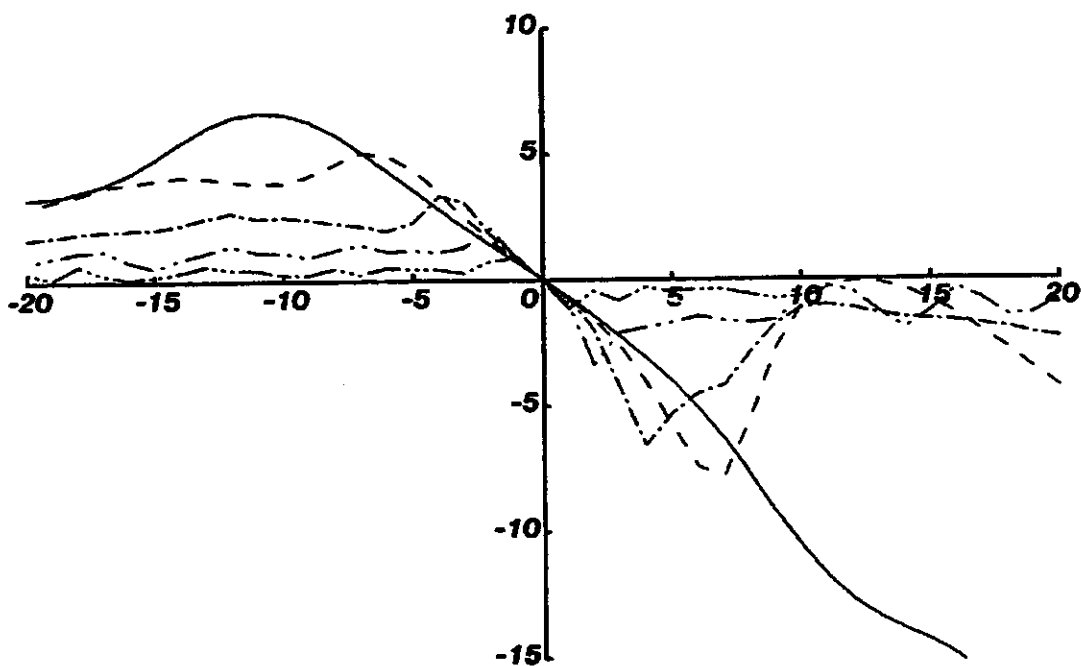
Figure 4-24. *House 2* pair, 10 *db* S/N, α_{TI}. See Figure 4-3.
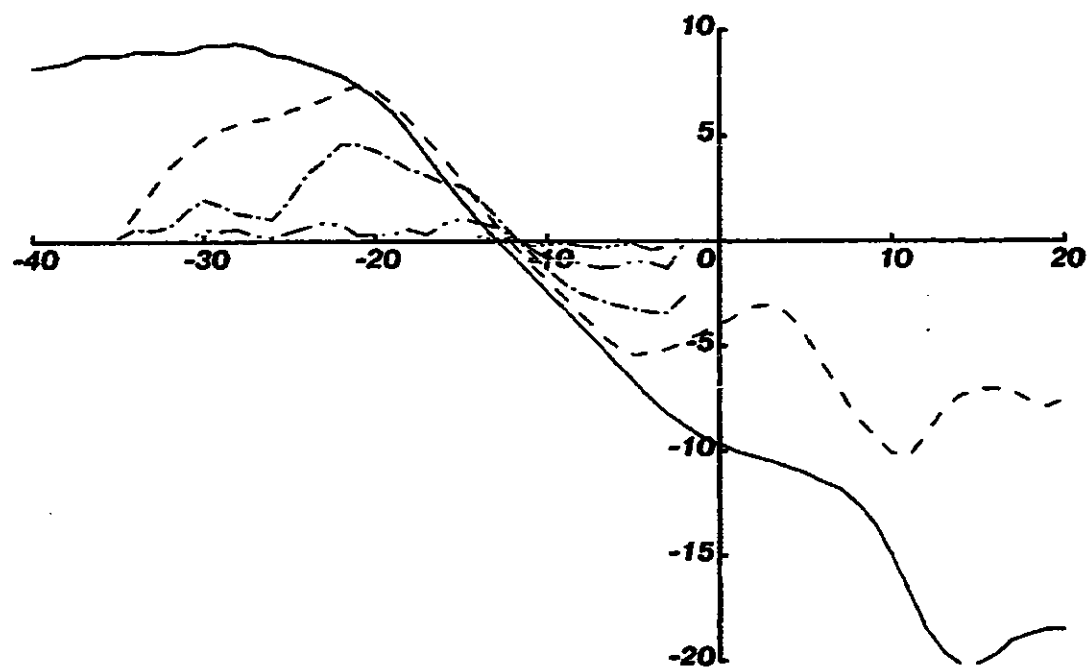
Figure 4-25. *House 2* pair, 0 *db* S/N, α_{PA}. See Figure 4-3.
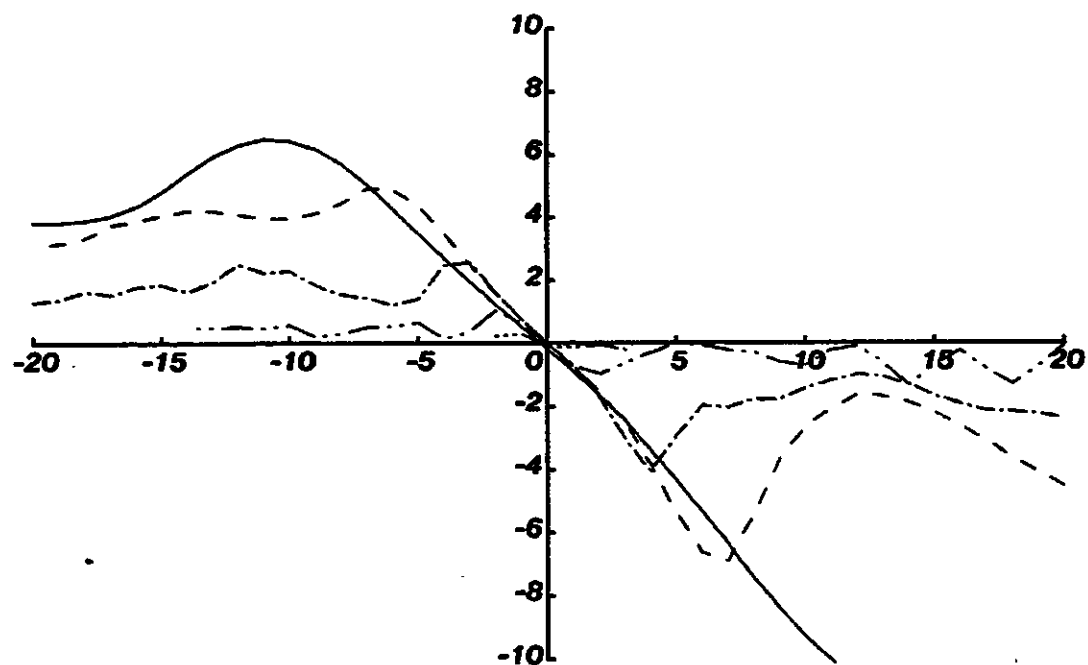
Figure 4-26. *House 2* pair, 0 *db* S/N, α_{TI}. See Figure 4-3.
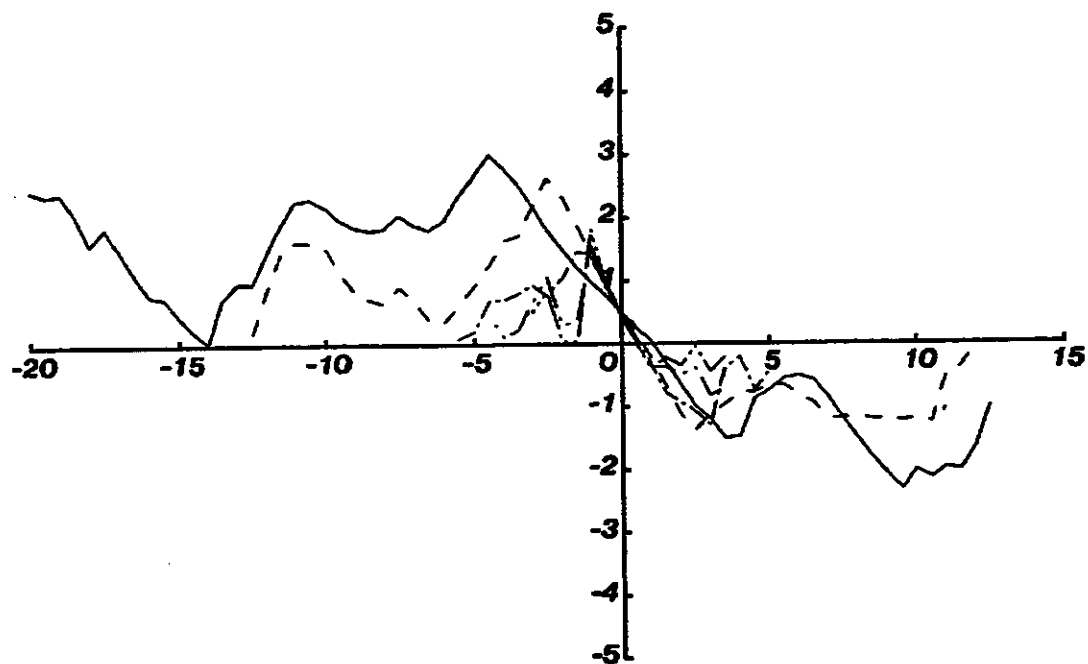
Figure 4-27. *Cart L* pair, α_{PA}. See Figure 4-3 for interpretation.
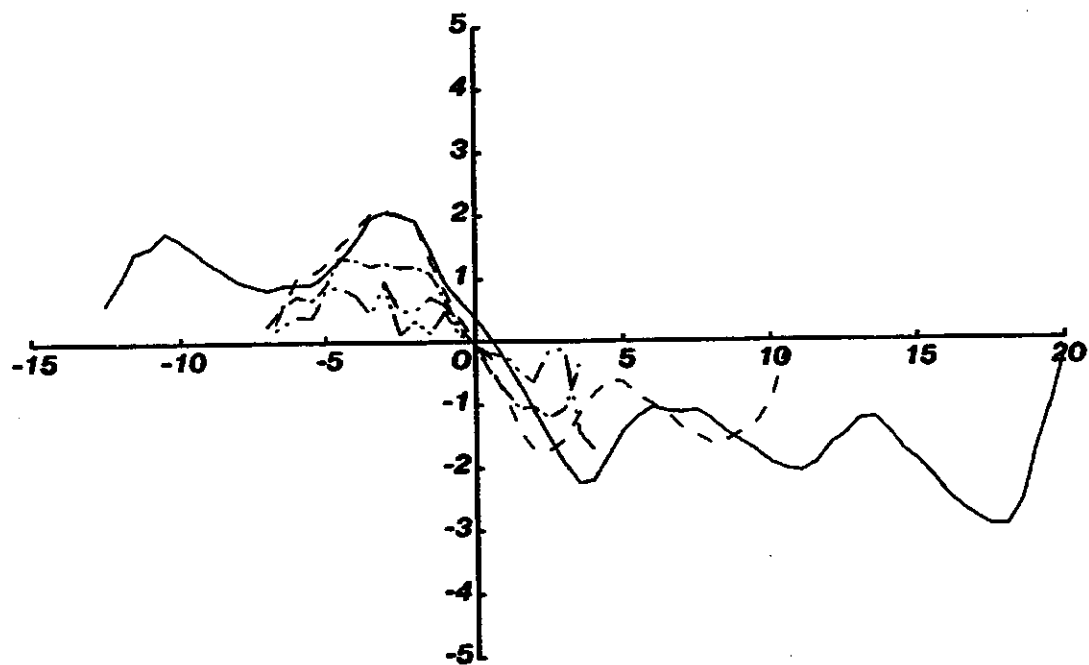
Figure 4-28. *Cart L* pair, α_{TI}. See Figure 4-3 for interpretation.
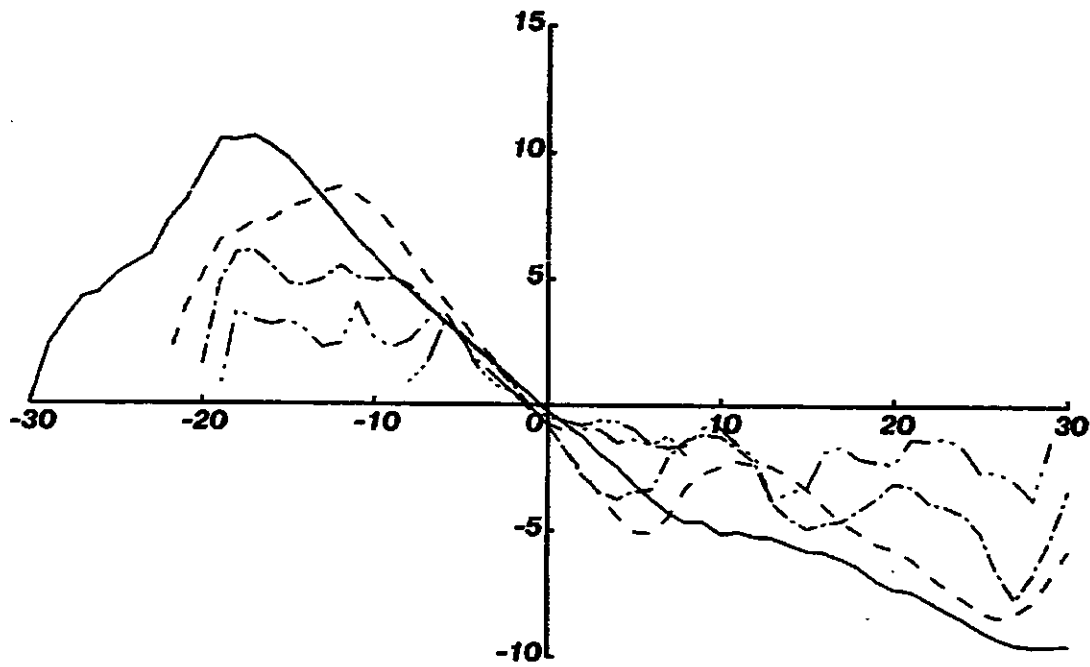
Figure 4-29. *Cart L* pair, α_{RO}. See Figure 4-3 for interpretation.

Figure 4-30. *Cart L* pair, r_x. See Figure 4-3 for interpretation.
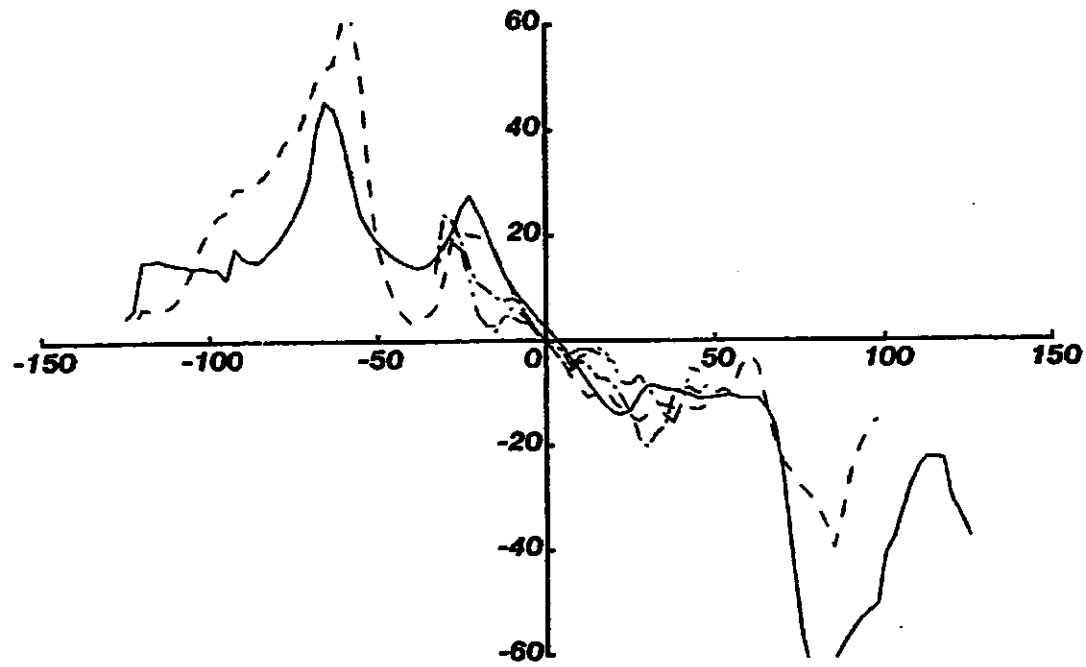
Figure 4-31. *Cart L* pair, r_y. See Figure 4-3 for interpretation.
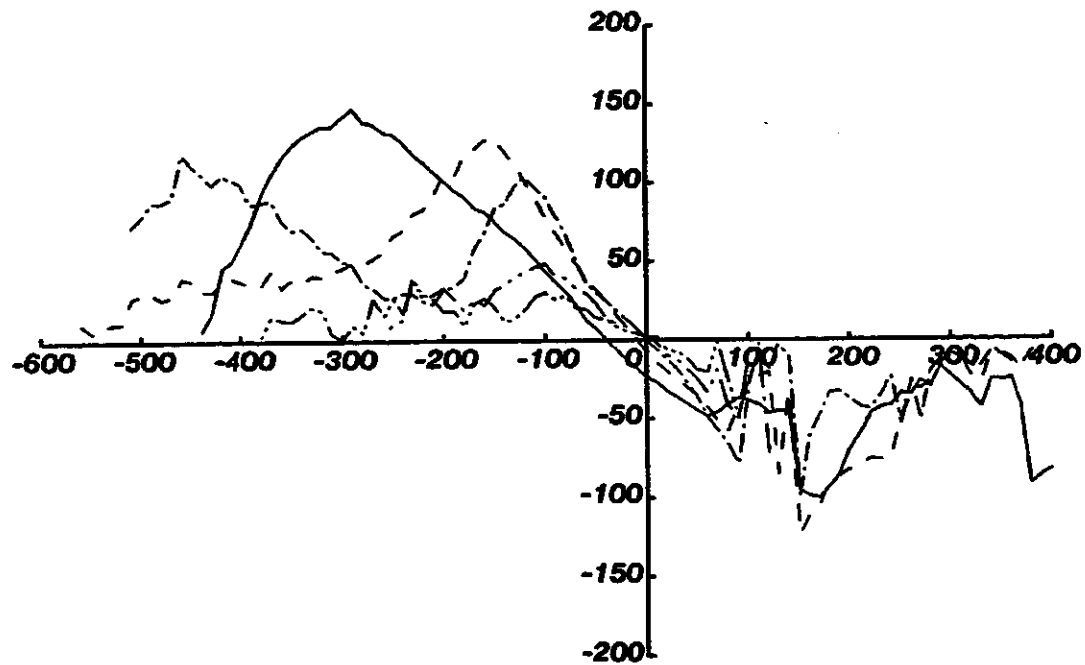
Figure 4-32. *Cart L* pair, r_z. See Figure 4-3 for interpretation.
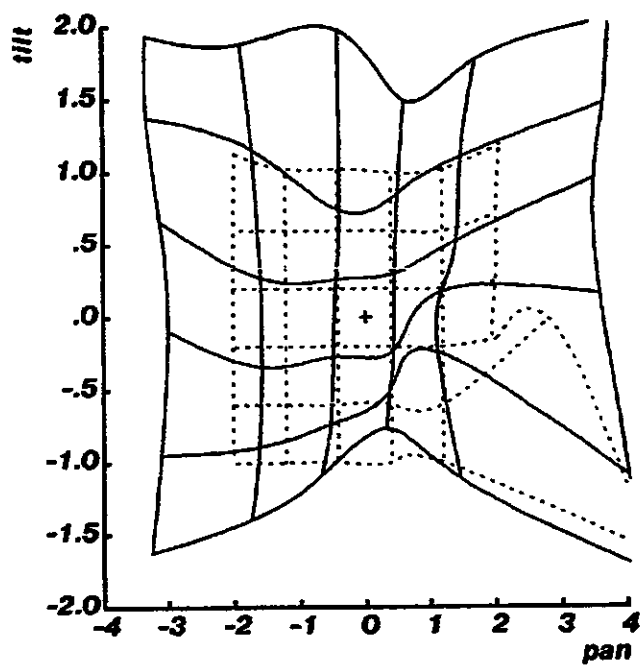
Figure 4-33. *House 1* pair, α_{PA} vs. α_{TI}.
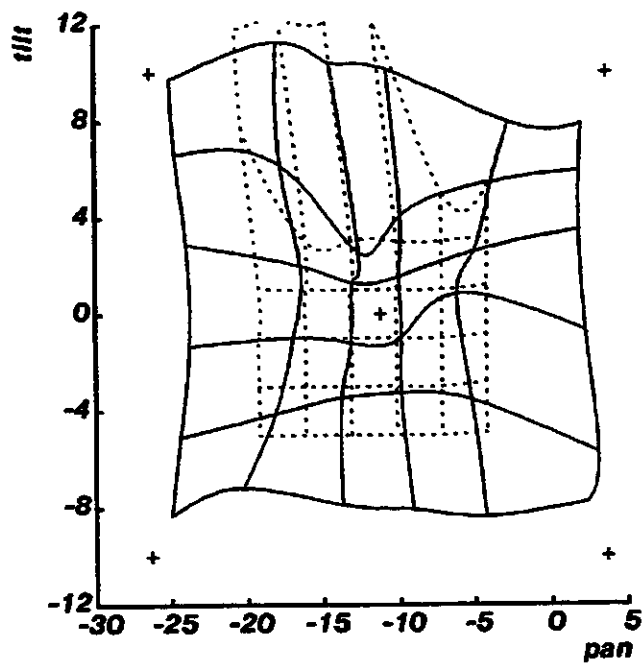
Figure 4-34. *House 2* pair, α_{PA} vs. α_{TI}.
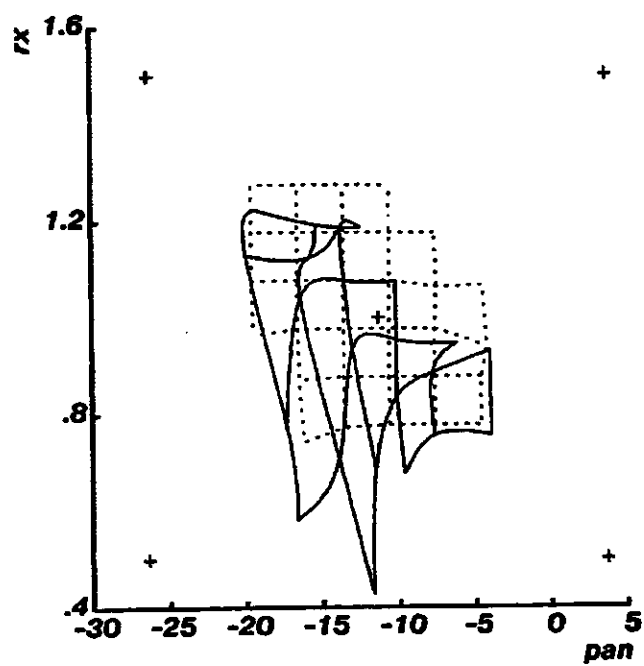
Figure 4-35. *House 2* pair, α_{PA} vs. r_x.
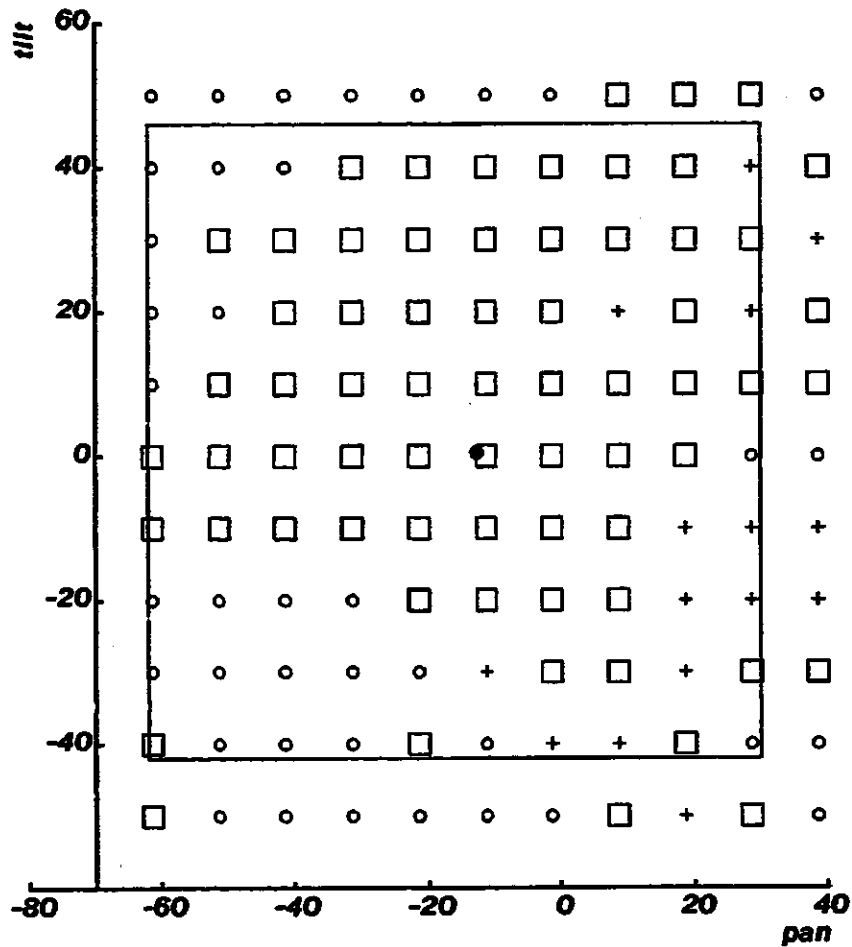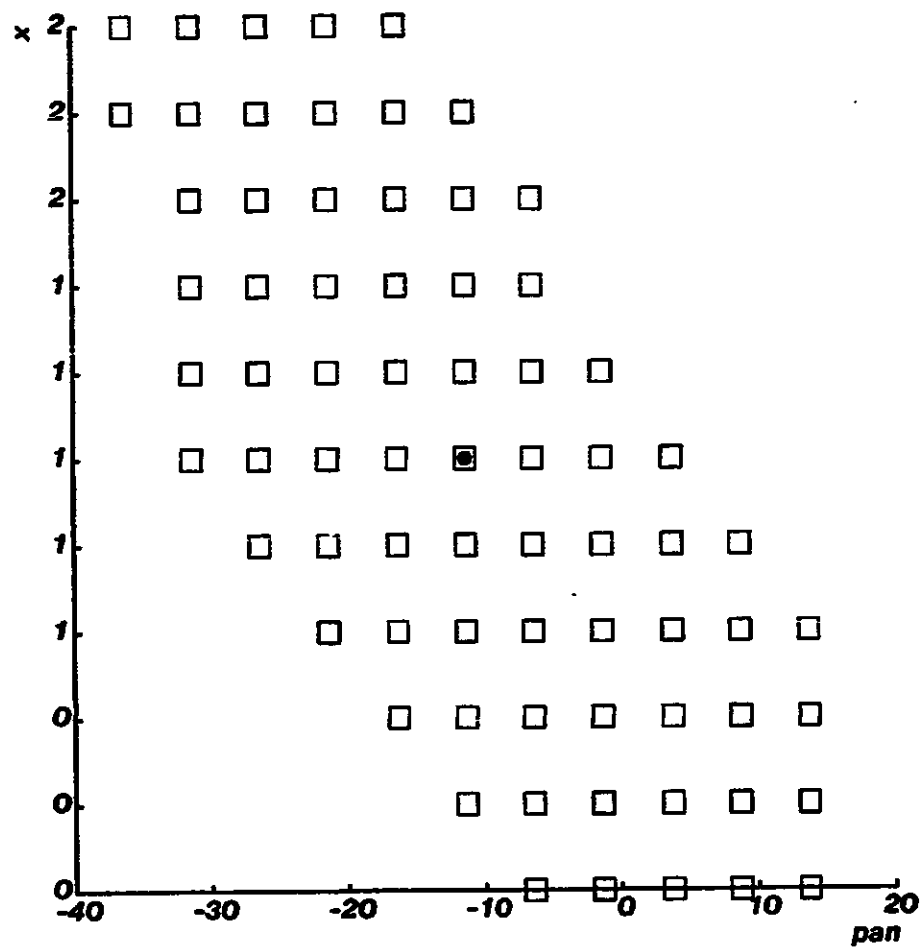
Figure 4-36. Convergence, α_{PA} vs. α_{TI}, 65 × 65 smoothing, *House 2* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: convergence value. Big box shows region predicted by the single parameters independently, as per Table 4-4.
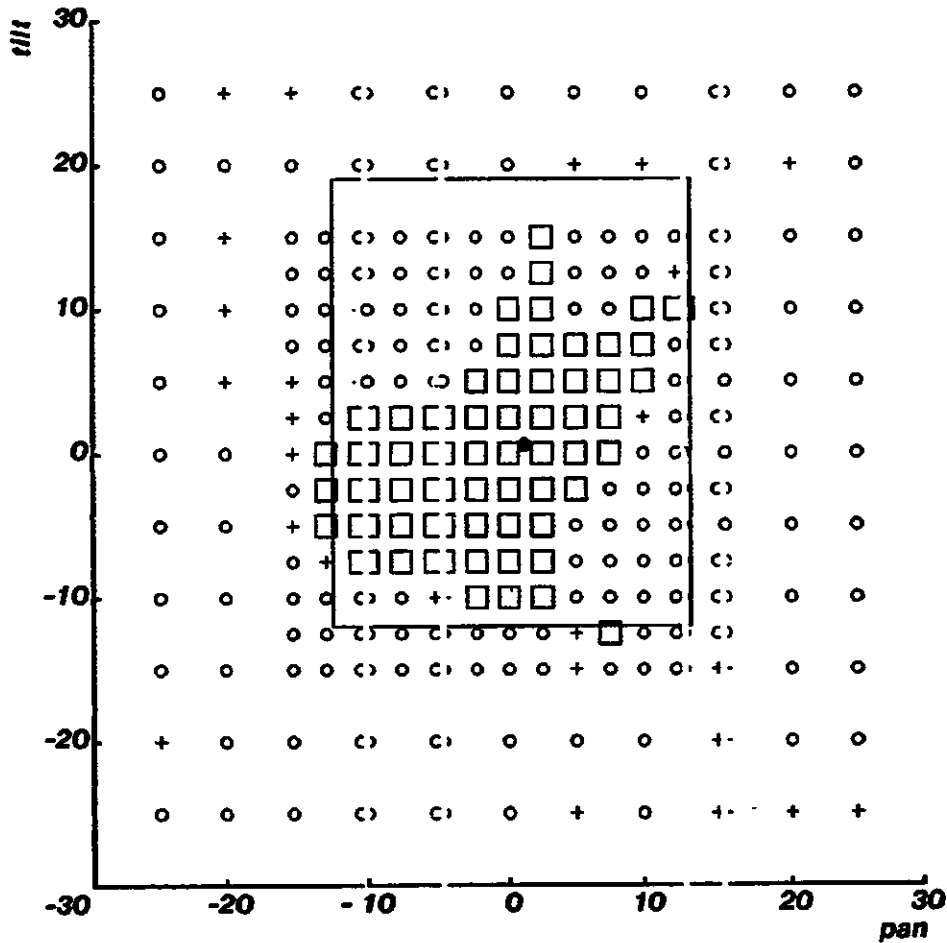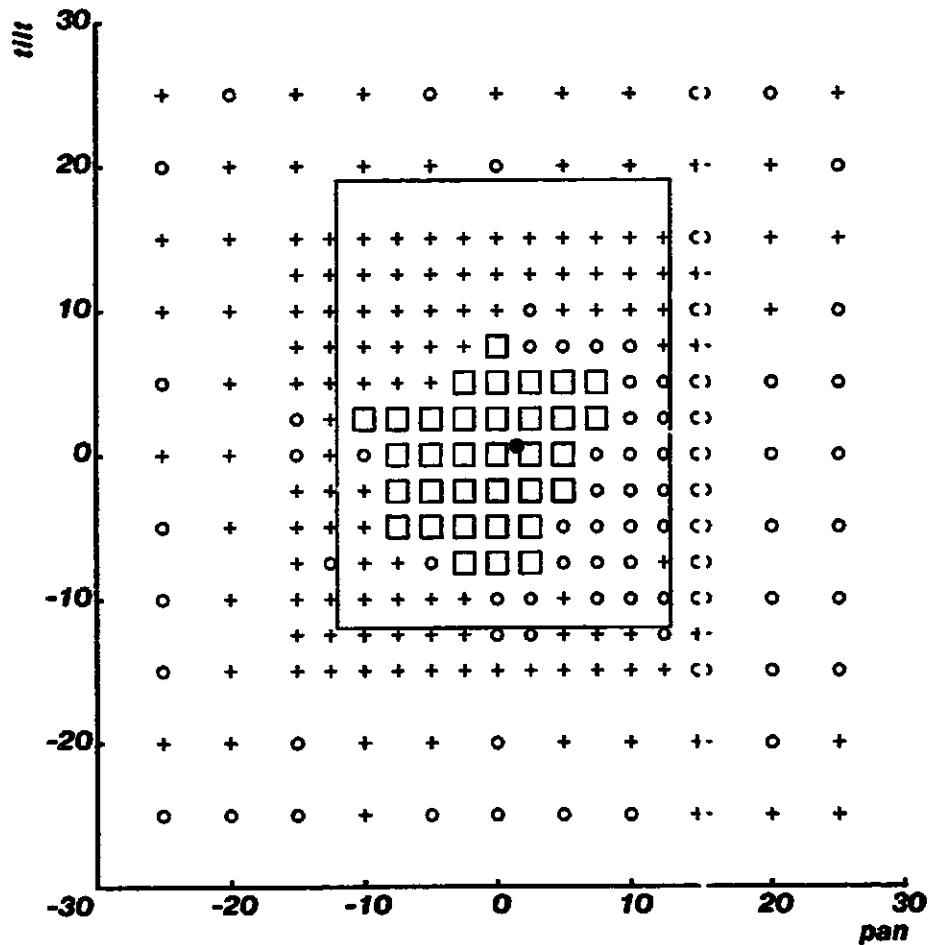
Figure 4-37. Convergence, α_{PA} vs. α_{TI}, 65 × 65 smoothing, *Cart L* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: convergence value. Big box shows region predicted by the single parameters independently, as per Table 4-4.
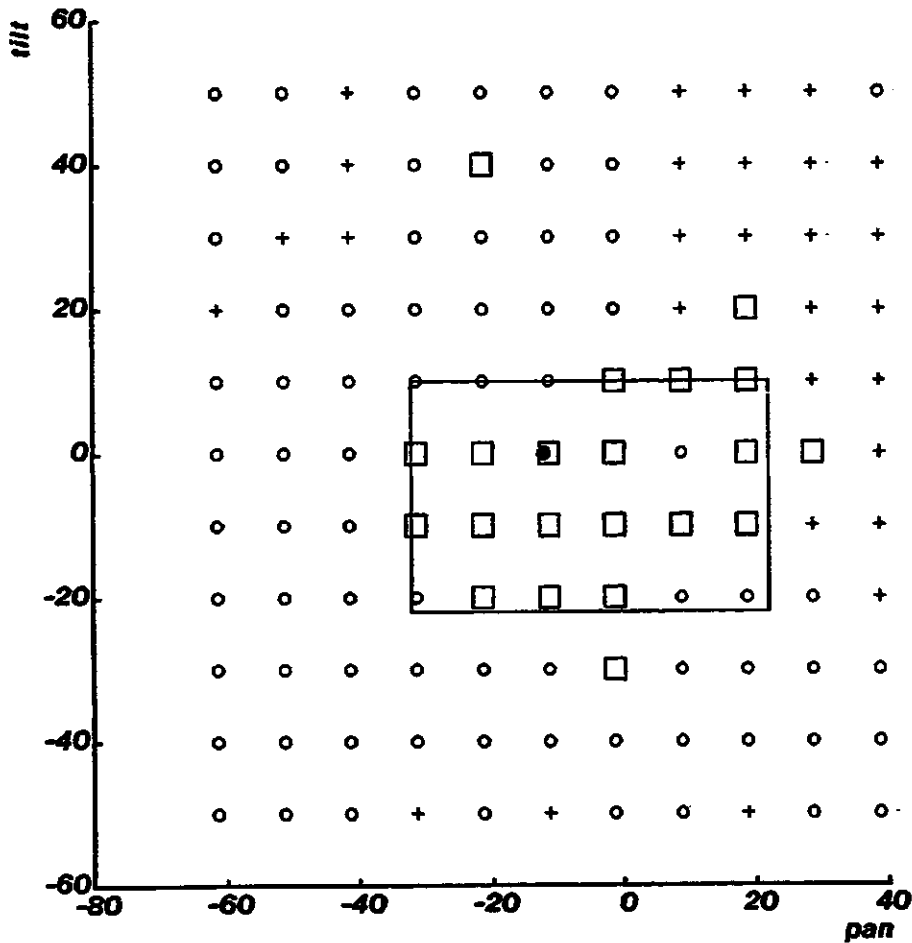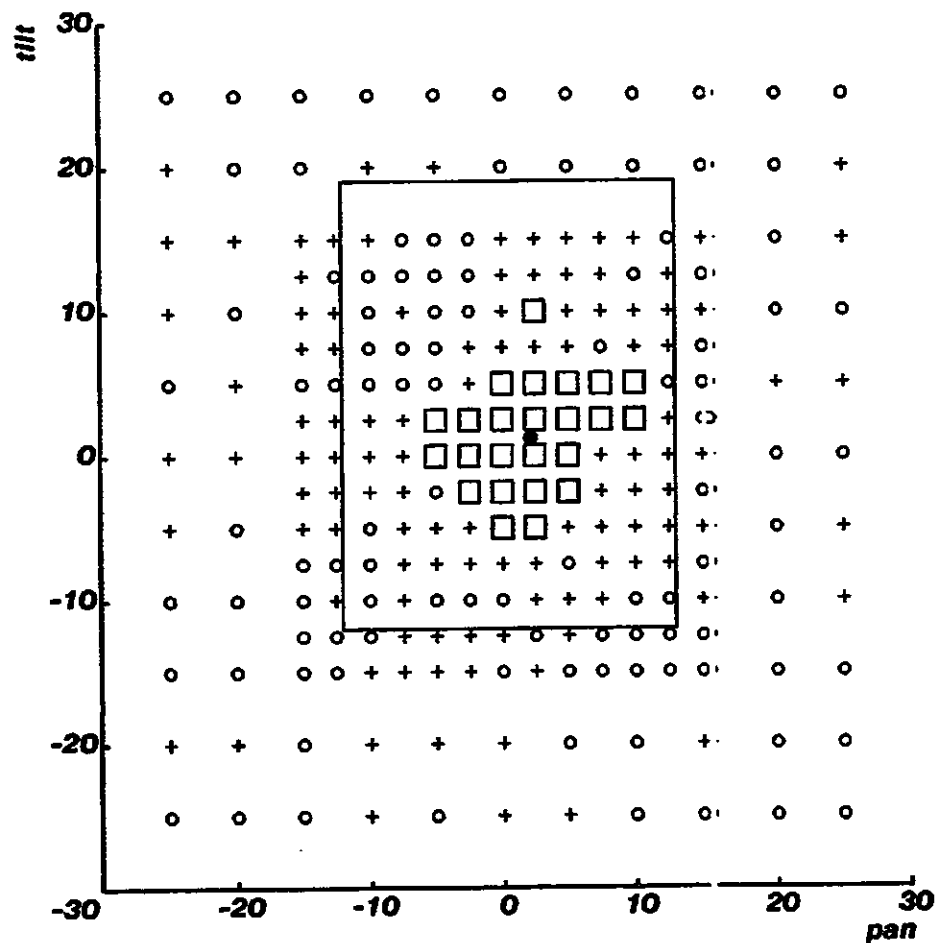
Figure 4–38. Convergence, α_{PA} vs. α_{TI}, 33 × 33 smoothing, *House 2* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: convergence value. Big box shows region predicted by the single parameters independently, as per Table 4–4.
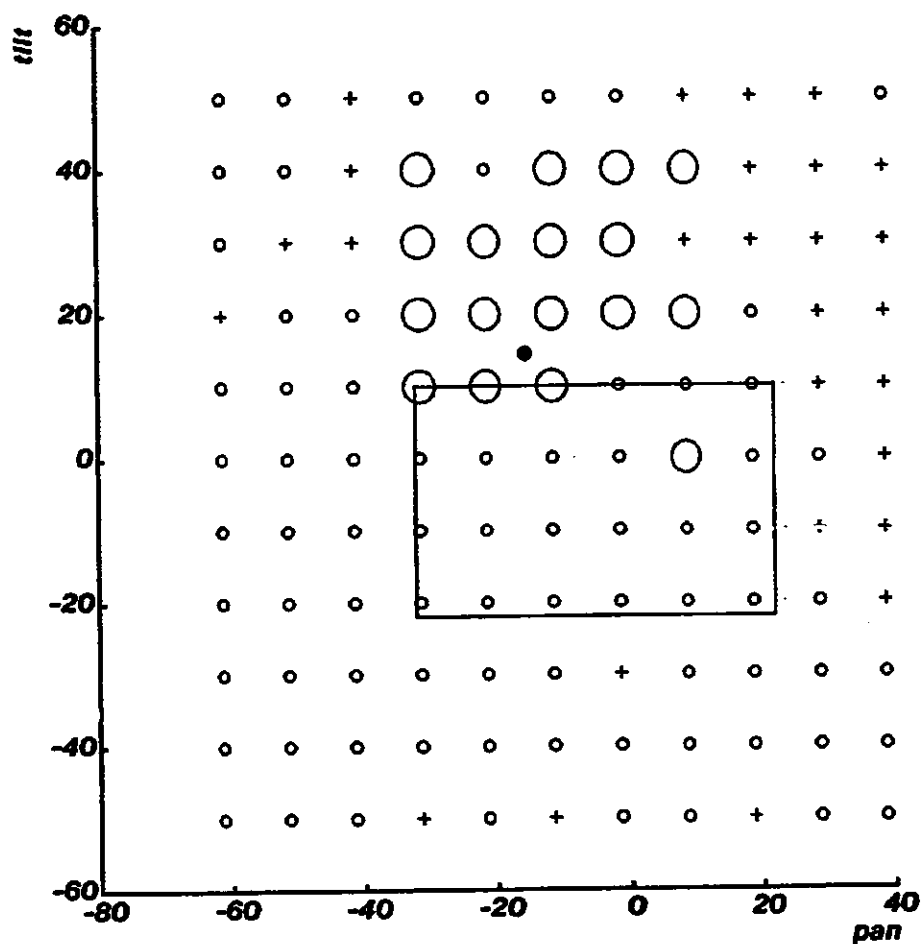
Figure 4-39. Convergence, α_{PA} vs. α_{TI}, 33×33 smoothing, *House 2* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: convergence value. Big box shows region predicted by the single parameters independently, as per Table 4-4.

Figure 4-40. Convergence, α_{PA} vs. r_z, 33 × 33 smoothing, *House 2* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: convergence value

Figure 4-41. Convergence, α_{PA} vs. α_{TI} with r_x sol ved for also, 65 × 65 smoothing, *Cart L* pair. Squares: convergence to right valu e; circles: convergence to wrong value; pluses: no convergence; dot: convergence value. Big box shows region predicted by the single parameters independently, as ;per Table 4-4.

Figure 4–42. Convergence, α_{PA} vs. α_{TI} with the other f our parameters solved for also, 65×65 smoothing, *Cart L* pair. Squares: convergence to right value; circles: convergence to wrong value; pluses: no convergence; dot: c onvergence value. Big box shows region predicted by the single parameters indepe ndently, as per Table 4–4.

α_{PA}	α_{TI}	α_{RO}	r_x	r_y	r_z	α_{PA}	α_{TI}	α_{RO}	r_x	r_y	r_z
-13.30	-2.000	-2.000	0.900	-0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	-2.000	0.900	-0.100	0.100	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-13.30	-2.000	-2.000	0.900	0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	-2.000	0.900	0.100	0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	-2.000	1.100	-0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	-2.000	1.100	-0.100	0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	-2.000	1.100	0.100	-0.100	-12.38	-0.437	-0.651	1.062	0.029	-0.059
-13.30	-2.000	-2.000	1.100	0.100	0.100	-12.38	-0.445	-0.655	1.061	0.030	-0.058
-13.30	-2.000	2.000	0.900	-0.100	-0.100	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-13.30	-2.000	2.000	0.900	-0.100	0.100	-12.40	-0.424	-0.642	1.062	0.029	-0.060
-13.30	-2.000	2.000	0.900	0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	-2.000	2.000	0.900	0.100	0.100	-12.41	-0.419	-0.637	1.063	0.029	-0.060
-13.30	-2.000	2.000	1.100	-0.100	-0.100	-12.39	-0.428	-0.645	1.062	0.029	-0.059
-13.30	-2.000	2.000	1.100	-0.100	0.100	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-13.30	-2.000	2.000	1.100	0.100	-0.100	-12.39	-0.440	-0.651	1.062	0.029	-0.059
-13.30	-2.000	2.000	1.100	0.100	0.100	-12.41	-0.418	-0.635	1.063	0.029	-0.060
-13.30	2.000	-2.000	0.900	-0.100	-0.100	-12.38	-0.445	-0.655	1.061	0.030	-0.058
-13.30	2.000	-2.000	0.900	-0.100	0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-13.30	2.000	-2.000	0.900	0.100	-0.100	-12.38	-0.483	-0.648	1.061	0.031	-0.058
-13.30	2.000	-2.000	0.900	0.100	0.100	-12.37	-0.531	-0.643	1.061	0.033	-0.058
-13.30	2.000	-2.000	1.100	-0.100	-0.100	-12.42	-0.405	-0.626	1.064	0.028	-0.060
-13.30	2.000	-2.000	1.100	-0.100	0.100	-12.37	-0.442	-0.655	1.061	0.029	-0.058
-13.30	2.000	-2.000	1.100	0.100	-0.100	-12.38	-0.446	-0.655	1.061	0.030	-0.059
-13.30	2.000	-2.000	1.100	0.100	0.100	-12.38	-0.481	-0.649	1.061	0.031	-0.058
-13.30	2.000	2.000	0.900	-0.100	-0.100	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-13.30	2.000	2.000	0.900	-0.100	0.100	-12.41	-0.416	-0.637	1.063	0.028	-0.060
-13.30	2.000	2.000	0.900	0.100	-0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-13.30	2.000	2.000	0.900	0.100	0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-13.30	2.000	2.000	1.100	-0.100	-0.100	-12.42	-0.389	-0.616	1.065	0.027	-0.062
-13.30	2.000	2.000	1.100	-0.100	0.100	-12.39	-0.432	-0.648	1.062	0.029	-0.059
-13.30	2.000	2.000	1.100	0.100	-0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-13.30	2.000	2.000	1.100	0.100	0.100	-12.43	-0.394	-0.618	1.065	0.028	-0.061
-9.30	-2.000	-2.000	0.900	-0.100	-0.100	-12.42	-0.393	-0.618	1.065	0.028	-0.062
-9.30	-2.000	-2.000	0.900	-0.100	0.100	-12.37	-0.442	-0.655	1.061	0.029	-0.058
-9.30	-2.000	-2.000	0.900	0.100	-0.100	-12.40	-0.420	-0.640	1.062	0.029	-0.059
-9.30	-2.000	-2.000	0.900	0.100	0.100	-12.38	-0.437	-0.651	1.062	0.029	-0.059
-9.30	-2.000	-2.000	1.100	-0.100	-0.100	-12.42	-0.383	-0.620	1.065	0.027	-0.063
-9.30	-2.000	-2.000	1.100	-0.100	0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-9.30	-2.000	-2.000	1.100	0.100	-0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-9.30	-2.000	-2.000	1.100	0.100	0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-9.30	-2.000	2.000	0.900	-0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-9.30	-2.000	2.000	0.900	-0.100	0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-9.30	-2.000	2.000	0.900	0.100	-0.100	-12.43	-0.397	-0.621	1.065	0.028	-0.061
-9.30	-2.000	2.000	0.900	0.100	0.100	-12.39	-0.432	-0.648	1.062	0.029	-0.059
-9.30	-2.000	2.000	1.100	-0.100	-0.100	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-9.30	-2.000	2.000	1.100	-0.100	0.100	-12.40	-0.436	-0.639	1.063	0.029	-0.060
-9.30	-2.000	2.000	1.100	0.100	-0.100	-12.43	-0.400	-0.623	1.065	0.028	-0.060
-9.30	-2.000	2.000	1.100	0.100	0.100	-12.43	-0.394	-0.618	1.065	0.028	-0.061
-9.30	2.000	-2.000	0.900	-0.100	-0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-9.30	2.000	-2.000	0.900	-0.100	0.100	-12.41	-0.415	-0.636	1.063	0.028	-0.060
-9.30	2.000	-2.000	0.900	0.100	-0.100	-12.40	-0.424	-0.642	1.062	0.029	-0.060
-9.30	2.000	-2.000	0.900	0.100	0.100	-12.40	-0.435	-0.640	1.063	0.029	-0.060
-9.30	2.000	-2.000	1.100	-0.100	-0.100	-12.42	-0.405	-0.626	1.064	0.028	-0.060
-9.30	2.000	-2.000	1.100	-0.100	0.100	-12.42	-0.405	-0.626	1.064	0.028	-0.060
-9.30	2.000	-2.000	1.100	0.100	-0.100	-12.41	-0.416	-0.637	1.063	0.028	-0.060
-9.30	2.000	-2.000	1.100	0.100	0.100	-12.38	-0.439	-0.653	1.061	0.029	-0.059
-9.30	2.000	2.000	0.900	-0.100	-0.100	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-9.30	2.000	2.000	0.900	-0.100	0.100	-12.42	-0.389	-0.616	1.065	0.027	-0.062
-9.30	2.000	2.000	0.900	0.100	-0.100	-12.43	-0.399	-0.622	1.065	0.028	-0.061
-9.30	2.000	2.000	0.900	0.100	0.100	-12.43	-0.399	-0.622	1.065	0.028	-0.061
-9.30	2.000	2.000	1.100	-0.100	-0.100	-12.43	-0.400	-0.623	1.065	0.028	-0.060
-9.30	2.000	2.000	1.100	-0.100	0.100	-12.43	-0.400	-0.623	1.065	0.028	-0.060
-9.30	2.000	2.000	1.100	0.100	-0.100	-10.89	41.519	27.068	1.743	-3.073	-0.836
-9.30	2.000	2.000	1.100	0.100	0.100	-12.42	-0.336	-0.635	1.064	0.025	-0.061

Table 4–43. Left columns are initial values of six parameters, right columns values after ten iterations. *House 2* pair, no noise.

α_{PA}	α_{TI}	α_{RO}	r_x	r_y	r_z	α_{PA}	α_{TI}	α_{RO}	r_x	r_y	r_z
-14.30	-3.000	-3.000	0.850	-0.150	-0.150	35.23	0.679	-12.511	-2.665	0.596	-0.041
-14.30	-3.000	-3.000	0.850	-0.150	0.150	-12.38	-0.442	-0.653	1.061	0.030	-0.059
-14.30	-3.000	-3.000	0.850	0.150	-0.150	-12.41	-0.416	-0.637	1.063	0.028	-0.060
-14.30	-3.000	-3.000	0.850	0.150	0.150	-12.40	-0.427	-0.642	1.062	0.029	-0.059
-14.30	-3.000	-3.000	1.150	-0.150	-0.150	-12.41	-0.418	-0.635	1.063	0.029	-0.060
-14.30	-3.000	-3.000	1.150	-0.150	0.150	-12.41	-0.419	-0.637	1.063	0.029	-0.060
-14.30	-3.000	-3.000	1.150	0.150	-0.150	-12.37	-0.442	-0.655	1.061	0.029	-0.058
-14.30	-3.000	-3.000	1.150	0.150	0.150	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-14.30	-3.000	3.000	0.850	-0.150	-0.150	-12.44	-0.056	-0.674	1.064	0.014	-0.061
-14.30	-3.000	3.000	0.850	-0.150	0.150	6.67	-19.424	-35.186	1.146	1.910	-1.272
-14.30	-3.000	3.000	0.850	0.150	-0.150	-12.39	-0.430	-0.644	1.062	0.029	-0.059
-14.30	-3.000	3.000	0.850	0.150	0.150	-12.40	-0.424	-0.642	1.062	0.029	-0.060
-14.30	-3.000	3.000	1.150	-0.150	-0.150	-12.38	-0.473	-0.646	1.061	0.031	-0.058
-14.30	-3.000	3.000	1.150	-0.150	0.150	-12.38	-0.481	-0.649	1.061	0.031	-0.058
-14.30	-3.000	3.000	1.150	0.150	-0.150	-12.37	-0.442	-0.655	1.061	0.029	-0.058
-14.30	-3.000	3.000	1.150	0.150	0.150	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-14.30	3.000	-3.000	0.850	-0.150	-0.150	-12.40	-0.427	-0.641	1.062	0.029	-0.060
-14.30	3.000	-3.000	0.850	-0.150	0.150	-12.40	-0.424	-0.642	1.062	0.029	-0.060
-14.30	3.000	-3.000	0.850	0.150	-0.150	6.49	-24.303	-25.886	-0.198	2.117	0.116
-14.30	3.000	-3.000	0.850	0.150	0.150	7.18	-24.142	-26.968	-0.181	1.992	0.185
-14.30	3.000	-3.000	1.150	-0.150	-0.150	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-14.30	3.000	-3.000	1.150	-0.150	0.150	-12.38	-0.445	-0.655	1.061	0.030	-0.058
-14.30	3.000	-3.000	1.150	0.150	-0.150	-12.38	-0.453	-0.651	1.061	0.030	-0.059
-14.30	3.000	-3.000	1.150	0.150	0.150	-12.40	-0.427	-0.643	1.063	0.029	-0.060
-14.30	3.000	3.000	0.850	-0.150	-0.150	-12.43	-0.400	-0.623	1.065	0.028	-0.060
-14.30	3.000	3.000	0.850	-0.150	0.150	-12.40	-0.424	-0.642	1.062	0.029	-0.060
-14.30	3.000	3.000	0.850	0.150	-0.150	6.10	-23.507	-25.470	-0.136	1.941	0.169
-14.30	3.000	3.000	0.850	0.150	0.150	-12.39	-0.716	-0.603	1.058	0.039	-0.048
-14.30	3.000	3.000	1.150	-0.150	-0.150	-12.42	-0.389	-0.616	1.065	0.027	-0.062
-14.30	3.000	3.000	1.150	-0.150	0.150	-12.41	-0.415	-0.636	1.063	0.028	-0.060
-14.30	3.000	3.000	1.150	0.150	-0.150	-12.43	-0.397	-0.622	1.064	0.028	-0.061
-14.30	3.000	3.000	1.150	0.150	0.150	-12.43	-0.276	-0.619	1.065	0.023	-0.064
-8.30	-3.000	-3.000	0.850	-0.150	-0.150	-12.42	-0.389	-0.616	1.065	0.027	-0.062
-8.30	-3.000	-3.000	0.850	-0.150	0.150	-12.38	-0.445	-0.655	1.061	0.030	-0.058
-8.30	-3.000	-3.000	0.850	0.150	-0.150	-12.40	-0.420	-0.640	1.062	0.029	-0.059
-8.30	-3.000	-3.000	0.850	0.150	0.150	-12.41	-0.418	-0.635	1.063	0.029	-0.060
-8.30	-3.000	-3.000	1.150	-0.150	-0.150	-12.43	-0.395	-0.621	1.065	0.028	-0.061
-8.30	-3.000	-3.000	1.150	-0.150	0.150	-12.41	-0.416	-0.637	1.063	0.028	-0.060
-8.30	-3.000	-3.000	1.150	0.150	-0.150	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-8.30	-3.000	-3.000	1.150	0.150	0.150	-12.50	-0.059	-0.537	1.067	0.019	-0.056
-8.30	-3.000	3.000	0.850	-0.150	-0.150	-12.37	-0.524	-0.640	1.061	0.033	-0.058
-8.30	-3.000	3.000	0.850	-0.150	0.150	-12.38	-0.478	-0.647	1.061	0.031	-0.058
-8.30	-3.000	3.000	0.850	0.150	-0.150	-12.42	-0.389	-0.616	1.065	0.027	-0.062
-8.30	-3.000	3.000	0.850	0.150	0.150	-12.41	-0.418	-0.635	1.063	0.029	-0.060
-8.30	-3.000	3.000	1.150	-0.150	-0.150	7.38	-24.257	-27.218	-0.193	2.004	0.187
-8.30	-3.000	3.000	1.150	-0.150	0.150	7.75	-24.505	-27.624	-0.217	2.030	0.192
-8.30	-3.000	3.000	1.150	0.150	-0.150	-12.39	-0.432	-0.646	1.062	0.029	-0.059
-8.30	-3.000	3.000	1.150	0.150	0.150	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-8.30	3.000	-3.000	0.850	-0.150	-0.150	-12.42	-0.405	-0.626	1.064	0.028	-0.060
-8.30	3.000	-3.000	0.850	-0.150	0.150	-12.37	-0.442	-0.655	1.061	0.029	-0.058
-8.30	3.000	-3.000	0.850	0.150	-0.150	-12.38	-0.455	-0.652	1.061	0.030	-0.059
-8.30	3.000	-3.000	0.850	0.150	0.150	-12.38	-0.224	-0.674	1.062	0.021	-0.061
-8.30	3.000	-3.000	1.150	-0.150	-0.150	-12.39	-0.427	-0.645	1.062	0.029	-0.059
-8.30	3.000	-3.000	1.150	-0.150	0.150	-12.38	-0.481	-0.649	1.061	0.031	-0.058
-8.30	3.000	-3.000	1.150	0.150	-0.150	-	-	-	-	-	-
-8.30	3.000	-3.000	1.150	0.150	0.150	-12.42	-0.405	-0.626	1.064	0.028	-0.060
-8.30	3.000	3.000	0.850	-0.150	-0.150	-12.43	-0.402	-0.624	1.064	0.028	-0.060
-8.30	3.000	3.000	0.850	-0.150	0.150	-12.43	-0.394	-0.618	1.065	0.028	-0.061
-8.30	3.000	3.000	0.850	0.150	-0.150	-12.39	-0.475	-0.642	1.061	0.031	-0.058
-8.30	3.000	3.000	0.850	0.150	0.150	-12.44	-0.390	-0.617	1.065	0.027	-0.061
-8.30	3.000	3.000	1.150	-0.150	-0.150	-12.43	-0.399	-0.622	1.065	0.028	-0.061
-8.30	3.000	3.000	1.150	-0.150	0.150	-12.38	-0.446	-0.655	1.061	0.030	-0.059
-8.30	3.000	3.000	1.150	0.150	-0.150	-12.44	-0.372	-0.629	1.064	0.027	-0.059
-8.30	3.000	3.000	1.150	0.150	0.150	-12.40	-0.555	-0.631	1.061	0.034	-0.057

Table 4–44. Left columns are initial values of six parameters, right columns values after ten iterations. *House 2* pair, no noise.

Chapter 5

Stereo:
Theory

5.1. Introduction

This chapter discusses in detail the theory behind the application of the method of differences to stereo vision. The most familiar example of stereo vision is of course the human visual system, in which binocular vision is used to determine the distance of objects from the viewer. Stereo vision requires matching points between two images, typically a left and a right image. But to avoid such bias, and to acknowledge the fact that points are chosen from one image and matched in the other, these images will be referred to as the *reference* and *test* images, respectively. In stereo vision, the known camera parameters c relating the reference and test cameras together with information about which points in the test image match various points p in the reference image are used to determine the distances $z(p)$ of the objects seen.

A review of the geometry of stereo vision is in order. Please refer to Figure 5–1. For any point p in the reference image, the object which generated the image at that point must lie at a three-space point q that is somewhere on the ray extending from the origin of the camera coordinate system through p. Thus the image of the same object in the test picture must lie somewhere on the projection of that ray onto the test picture (line CD in the figure). This line in the test picture is known as the *epipolar line* corresponding to the point p in the reference picture. The epipolar line is determined purely by the geometry of the cameras and is independent of the scene being viewed. Furthermore, consider the family of planes that contain the origins of the two camera systems; each such plane intersects each picture plane in a line. Each such pair of lines (for example, AB and CD in the figure) are known as *corresponding epipolar lines*, because each point on an epipolar line in either image finds its match on the corresponding epipolar line in the other image. Of course this family of planes covers all of three-space and the two families of epipolar lines thereby generated cover all of each image, so the preceding statement applies to any point in either image. Thus the stereo matching problem is really a family of one-dimensional matching problems,
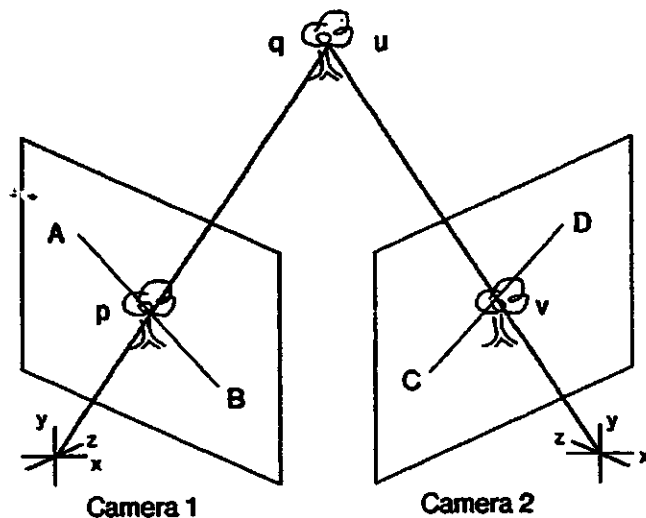
Figure 5–1. Camera model for stereo. Camera 1 defines the *reference* picture and coordinate system, with the origin at the "pinhole." Camera 2 defines the *test* picture and coordinate system. For any point $p = [p_x \quad p_y]$ in the image there is a point $q = [q_x \quad q_y \quad q_z]$ in three-space which produced the image, at depth $z(p) = q_z$. The point q lies somewhere on the ray from the origin of the Camera 1 coordinate system through the point p. Its matching point $v = [v_x \quad v_y]$ lies on the corresponding epipolar line CD. Lines AB and CD are corresponding epipolar lines, as discussed in the text.

although to treat it as a family of *independent* one-dimensional matching problems would be to ignore valuable constraints between the matchings for neighboring epipolar lines.

Nevertheless, stereo is essentially a problem of determining one parameter, the distance (or depth) $z(p)$, at each of some set of points p. Stereo problems fall into two (ill-defined) classes: the set of points p at which the distance is to be determined may be either sparse or dense. The sparse case might arise for example when a number of points, for example points near edges or corners, have been picked out of the reference image by an interest operator as useful to a higher level process and likely to be unambiguously matched in the test image. The dense case typically involves finding a depth map, that is the depth at every point in an image, or at least at every point which is not occluded. The mathematics of these two cases is essentially identical, but the implementation is different, as discussed below.

A special case of the camera geometry occurs when the cameras are fully parallel, i.e. when the coordinate system of the test camera is related to the coordinate system of the reference camera by a simple translation in the x direction. In this case, the epipolar lines correspond to scan lines. This is a considerable simplification of the geometry, and allows the

algorithms to be implemented more efficiently. Under carefully controlled conditions, this condition can be met. Where the cameras were not in fact parallel, a resampling technique can be used to produce a stereo pair of the same scene viewed from parallel cameras. This was in fact done with the natural images used in the next chapter. While this may or may not be advantageous in a real application, investing the time to resample the images pays off in increased speed in running the subsequent experiments. Thus, all the mathematics presented in subsequent sections assumes the cameras are parallel. The non-parallel case requires straightforward but tedious modifications of the following equations.

Using parallel cameras allows the distance $z(\mathbf{p})$ to be expressed naturally as a disparity $h(\mathbf{p})$. This disparity is closely related to the disparity definition used earlier because it measures the distance on the image plane in (say) pixels between the object's actual position and some reference position. In particular, it measures the distance between the object's actual position in the test picture and the position it would be at in the test picture if the object were moved away from the reference camera origin to infinity. Referring to Figure 5-1, the disparity measures the position of the object on the epipolar line CD as the object is moved along the ray proceeding from the camera 1 origin. The disparity is 0 for objects at infinity and gets larger as the object gets nearer. There is of course a one-to-one relationship between the disparity and the distance. For parallel cameras, the disparity representation is particularly natural because it measures the distance along a scan line between the object's position in the reference image and its position in the test image, since for parallel cameras an object at infinity appears at the same position in both the reference and test views. Furthermore, representing the distance as a disparity rather than a z value makes the best use of limited precision available for representing the distance. This is because the accuracy to which the distance can be determined is ultimately limited by the pixel size; the absolute error in the z value estimate for a match error of, say, 1 pixel is larger for large z values than for small ones, so representing the distance as a z value wastes precision at far distances. Using the disparity representation avoids this problem.

Finally, all of the algorithms that will be presented in this chapter are, like the navigation algorithm, iterative. This means that one starts out with a field of disparity estimates $h_0(\mathbf{p})$, and at each step computes a field of increments $\Delta h_i(\mathbf{p})$, so that

$$h_{i+1}(\mathbf{p}) = h_i(\mathbf{p}) + \Delta h_i(\mathbf{p}).$$

The algorithms that described here will be concerned with computing the increments $\Delta h_i(\mathbf{p})$. In each case we will assume that we have a set of current disparity estimates $h(\mathbf{p})$ and we wish to compute the increments $\Delta h(\mathbf{p})$. For convenience, let us define

$$\mathbf{h}(\mathbf{p}) = [h(\mathbf{p}) \quad 0].$$

This is just the disparity represented as a vector in the picture plane, which allows us to write such things as $\mathbf{p} + \mathbf{h}(\mathbf{p})$ conveniently.

5.2. Single-point algorithm

The simplest algorithm is the one-point algorithm mentioned in Section 4.2 for adjusting the $z(p)$ values in the optical navigation experiments. Expressed in disparity notation, it is

$$\Delta h(p) = \frac{I_2(p + h(p)) - I_1(p)}{D_x I_2(p + h(p))}.$$

This of course suffers from the problems of any single-point algorithm, namely that it depends very heavily on the linearity of the image intensities, and that it incorporates information from only a single pixel in each disparity estimate. It was used in the previous chapter for adjusting the z values for four reasons: First, in that case the z estimates were presumably very accurate to start with, which means that the disparity error was smaller and so the linearity assumption was less important. Second, because of the way they were chosen, the reference points were in regions of large intensity gradient, and so the linearity assumption tended to be accurate. Third, because the points were in regions of high intensity gradient, a given photometric error results in less geometric error; this is just another way of saying that edges provide more accurate matches. Finally, using a method that is sensitive to problems that the method of differences might encounter made it possible to weed out those points that were not robust with respect to the method. That is, by monitoring whether each point did well in this simple version of the method of differences, it was possible to eliminate those points that would not provide good information in a more complex version of the method of differences, namely the navigation algorithm.

5.3. Average and least-squares algorithms

An averaging algorithm analogous to that in equation (2-7) is possible. The corresponding equation, modified for stereo, is

$$\Delta h(p) = \sum_{p' \text{ near } p} \frac{I_2(p' + h(p')) - I_1(p')}{D_x I_2(p' + h(p'))}. \tag{5-1}$$

This is presented here for completeness and for comparison with the least-squares algorithm, which is the basis for all the experiments reported here.

The least-squares equation is derived from the formula for the error that is to be minimized at every point p:

$$E_p = \sum_{p' \text{ near } p} \left(I_2(p' + h(p') + \Delta h(p)) - I_1(p') \right)^2. \tag{5-2}$$

That is, given a disparity field $h(p)$, we wish to calculate a change $\Delta h(p)$. This change should be such that, for each point p, adding the change $\Delta h(p)$ at each point p' in a neighborhood of p minimizes the error over that neighborhood. This formulation has two important consequences. First, requiring that the change in disparity at p be arrived at by

asking each point $\mathbf{p}'$ in the neighborhood of $\mathbf{p}$ what it thinks about that change imposes a sort of smoothness constraint on the change, and thus on the disparity field. This is an implicit smoothness constraint, to be contrasted with the explicit smoothness constraint used by Horn & Schunck (1981). Second, this formulation results in a very efficient implementation in the case of a dense field of disparities and a rectangular neighborhood, as discussed below.

The error $E_\mathbf{p}$ from equation (5-2) is minimized by using the usual linear approximation,

$$I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}') + \Delta\mathbf{h}(\mathbf{p})) \approx I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}')) + \Delta\mathbf{h}(\mathbf{p})D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}')) \qquad (5\text{-}3)$$

Note that $D_x I_2$ is just the x component of the intensity gradient of the test image I_2; only the x component is used because the epipolar lines correspond to scan lines. Plugging this into equation (5-2) and minimizing yields

$$\Delta h(\mathbf{p}) = \frac{\displaystyle\sum_{\mathbf{p}' \text{ near } \mathbf{p}} (I_1(\mathbf{p}') - I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}')))\, D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}'))}{\displaystyle\sum_{\mathbf{p}' \text{ near } \mathbf{p}} D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}'))^2}. \qquad (5\text{-}4)$$

This equation shows even more clearly the implicit smoothness constraint: the change in disparity is computed as the ratio of two quantities each of which is essentially an "image" smoothed over the neighborhood defined by "near."

5.4. Weighting

As mentioned in Chapter 2, equation (5-4) can in fact be considered a weighted version of (5-1), where each term in the sum in (5-1) is multiplied by a weighting function $w(\mathbf{p}')$, in this case given by

$$w(\mathbf{p}') = D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}'))^2, \qquad (5\text{-}5)$$

and then the result is divided by the sum of the weights $\sum_{\mathbf{p}' \text{ near } \mathbf{p}} w(\mathbf{p}')$. In this case the weighting function serves to give more credibility to those points where the gradient is large, which are points near edges from which the most information can be obtained.

A more general weighting formulation is possible; this was presented in equation (2-14) on page 20. Reformulated for the stereo case, this equation becomes

$$\Delta h(\mathbf{p}) = \frac{\displaystyle\sum_{\mathbf{p}' \text{ near } \mathbf{p}} (I_1(\mathbf{p}') - I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}')))\, D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}'))w(\mathbf{p}')}{\displaystyle\sum_{\mathbf{p}' \text{ near } \mathbf{p}} D_x I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}'))^2 w(\mathbf{p}')}. \qquad (5\text{-}6)$$

One subtle point to notice is that the $w(\mathbf{p}')$ in (5-5) might more properly have been written as $w(\mathbf{p}' + \mathbf{h}(\mathbf{p}))$, because it is a property of the point $\mathbf{p}' + \mathbf{h}(\mathbf{p})$ that we are trying to match against in the test image. However, it is not written in this form in (5-6), so as to leave its form less constrained. For example, it may depend on the difference between the image intensity gradients at $\mathbf{p}'$ in the reference image and at $\mathbf{p}' + \mathbf{h}(\mathbf{p}')$ in the right image. Some examples of such weighting functions, among others, are given in starting at equation (2-15) on page 20.

5.5. Stability

The question of the stability of algorithms for stereo analysis derived from the method of differences is much more easily resolved than for optical navigation. This is because the method of differences involves only one division at each point in the image; thus the question of stability revolves around the size of the denominator,

$$\sum_{\mathbf{p'} \text{ near } \mathbf{p}} D_x I_2(\mathbf{p'} + \mathbf{h}(\mathbf{p'}))^2. \tag{5-7}$$

If this denominator is zero, the result is undefined; if the denominator is small, the result is unstable, in that small amounts of noise will cause large changes in the answer. The denominator in turn is small if the neighborhood near the point $\mathbf{p} + \mathbf{h}(\mathbf{p})$ in the test image that matches the point $\mathbf{p}$ in the reference image is "bland", that is nearly uniform. But since we expect the neighborhood around $\mathbf{p} + \mathbf{h}(\mathbf{p})$ in the test image to be very similar to the neighborhood around $\mathbf{p}$ in the reference image (indeed, this supposition is the entire basis of the algorithm), this is equivalent to the whether the region around $\mathbf{p}$ is uniform. This supports the intuitive idea that matching is best done in the neighborhood of edges, corners, etc. Indeed, the match for a bland region is inherently ambiguous, and no technique can be expected to do well in this case.

This also suggests that we can detect whether the method of differences is giving us good results in matching. The idea is to take the size of the denominator in (5-7) as an indicator of the reliability of the match. However, a more general formulation of this idea is possible. As discussed in the previous section, (5-4) can in fact be considered a weighted average algorithm, with the weight given by $w(\mathbf{p'}) = D_x I_2(\mathbf{p'} + \mathbf{h}(\mathbf{p'}))^2$. Thus, the denominator proposed as a measure of the reliability is just the sum of the weights in the neighborhood of $\mathbf{p}$! This provides us with another line of reasoning confirming that size of the denominator represents the reliability of the result.

Interestingly, information about whether an image is being matched well is in itself useful to higher level processes. Mismatch between two images constitutes evidence of occlusion or of specular reflections. For example, knowledge of occlusions provides evidence for the three-dimensional structure of an object. Furthermore, humans observing stereograms of scenes with shiny surfaces report that those surfaces have a shiny appearance just from the cue provided by the difference in intensities perceived by the two eyes for the surfaces. Thus, the reliability measure proposed here can provide additional information about the three-dimensional structure of a scene and about the surface properties of the objects in the scene.

5.6. Solving for brightness and contrast

These results might be improved in some cases by including in the camera model some photometric parameters as well as the geometric ones. This is because two cameras viewing the same scene, or the same camera viewing the same scene at two different times, will

not necessarily report the same intensity values for each corresponding point. This error has two sources: first, the cameras will not necessarily produce the same response to the same input; second, the specular component of reflection can cause an object to have a different apparent brightness from different viewpoints. This can result in not only the familiar specular reflections, but also in more subtle intensity differences. While it seems feasible to control the geometric parameters relating the two cameras adequately so that the epipolar lines correspond to scan lines, similarly controlling the photometric parameters so that equal intensities produce equal pixel values seems impractical. Moreover, this does not address the problem of specular reflections.

This leaves the question of how to model the intensity differences. Since we want to be able to handle specular reflections, we need a field of parameters rather than global parameters. But individual pixels, or small collections of pixels, generally lie entirely within a region or lie at the border of two regions. This means that a model with two parameters will be able to locally model any image intensity transformation. This naturally leads to the choice of the two parameters $\beta(\mathbf{p})$ (brightness, bias) and $\gamma(\mathbf{p})$ (contrast, gain). The camera model now becomes

$$\gamma(\mathbf{p})I_1(\mathbf{p}) + \beta(\mathbf{p}) = I_2(\mathbf{p} + \mathbf{h}(\mathbf{p})).$$

The intention is that while the β and γ fields vary with each pixel, they will in general do so smoothly because they will be calculated using information from a neighborhood of pixels. This is in much the same way that the computed disparity fields are expected to vary smoothly, due to the implicit smoothness constraint, as discussed above.

The error after a change $\Delta\mathbf{h}(\mathbf{p})$ in the disparity field $\mathbf{h}(\mathbf{p})$, corresponding to equation (5-2), is now

$$E = \left(\sum_{\mathbf{p}' \text{ near } \mathbf{p}} I_2(\mathbf{p}' + \mathbf{h}(\mathbf{p}') + \Delta\mathbf{h}(\mathbf{p})) - \gamma(\mathbf{p})I_1(\mathbf{p}') - \beta(\mathbf{p}) \right)^2.$$

This is solved by again using the linear approximation of equation (5-3), and minimizing with respect to $\Delta\mathbf{h}(\mathbf{p})$, $\beta(\mathbf{p})$, and $\gamma(\mathbf{p})$; this yields three linear equations in the three unknowns for each point $\mathbf{p}$:

$$\beta\sum I_2' + \gamma\sum I_1 I_2' - \Delta h\sum (I_2')^2 = \sum I_2 I_2',$$

$$\beta\sum I_1 + \gamma\sum I_1^2 \ - \Delta h\sum I_2' I_1 = \sum I_2 I_1, \qquad (5\text{-}8)$$

$$\beta\sum 1 + \gamma\sum I_1 \ - \Delta h\sum I_2' = \sum I_2.$$

Here the obvious abbreviations are used. The solution is straightforward.

5.7. Implementation

It may seem expensive to calculate the sums in (5-6) or (5-8) for every point **p** of an image, because these sums are over potentially large neighborhoods and because they must be computed for every point in the image. However, given the right definition of "near" in these sums, the stereo map can be calculated very efficiently, in time essentially independent of the size of the neighborhood. The definition of "near" that we have in mind is that of a rectangular neighborhood around **p**; that is **p'** is "near" **p** if

$$\mathbf{p}_x - \delta_x \le \mathbf{p}'_x \le \mathbf{p}_x + \delta_x, \quad \text{and}$$

$$\mathbf{p}_y - \delta_y \le \mathbf{p}'_y \le \mathbf{p}_y + \delta_y,$$

for a given δ_x and δ_y. Now computation of equation (5-6) amounts to uniform smoothing of the "images"

$$(I_1(\mathbf{p}) - I_2(\mathbf{p} + \mathbf{h}(\mathbf{p}))) D_x I_2(\mathbf{p} + \mathbf{h}(\mathbf{p})) w(\mathbf{p}), \quad \text{and}$$

$$D_x I_2(\mathbf{p} + \mathbf{h}(\mathbf{p}))^2 w(\mathbf{p}),$$

over a rectangular neighborhood. But by a well-known technique, reviewed in Appendix B, this can be accomplished in time essentially independent of the size of the neighborhood.

This optimization applies only to calculating a disparity for each point **p**. If we are calculating a disparity only for a small number of points, the incremental smoothing technique is not applicable. However, as the number of points increases, a point should be reached where it would be better to use the technique described above to compute a full disparity map, even if we are not interested in its value at all points.

5.8. Summary

This chapter has shown how the method of differences can be used to compute a disparity map. This primary contribution of this chapter was to show how, by careful design of the algorithm, we can allow each point to include information from a large neighborhood of points in running time which is essentially independent of the size of the neighborhood. The method requires about 30 sec per iteration for a 250 × 250 image on a VAX 11/780 (independent of smoothing time). Its regular structure makes it quite suitable for implementation on special-purpose hardware. The resulting algorithm incorporates an implicit smoothness constraint. That is, the smoothness constraint was not explicitly formulated as it is in the Horn & Schunck (1981), but the resulting algorithm constrains the disparity to vary smoothly nonetheless. Not surprisingly, numerical stability considerations show that the algorithm should do better in the presence of detail than it does in the bland portions of the image.

This concludes the theoretical development necessary to apply the method of differences to the computation of a stereo disparity map. The next chapter presents results from experiments using this algorithm.

Chapter 6

Stereo:
Experiments

6.1. Introduction

This chapter presents the results of experiments testing the stereo algorithms presented in the last chapter. These experiments were conducted on both synthetic and real data.

The synthetic data consisted of random dot stereograms. The random-dot stereograms were constructed as follows. First, generate the image $I_2(p)$ whose pixel values are assigned independent uniformly distributed random numbers ranging from 0 to 255. This serves as the right image of a stereo pair. Next, construct a disparity map $h(p)$ by some method, as described below. Finally, the left image, I_1, is assigned pixel values according to the formula $I_1(p) = I_2(p+h(p))$. Since the disparity can be a fractional quantity, this last step may require interpolation; linear interpolation suffices. The disparity values are actually represented to eight bits of precision. The images have 250 rows and 250 columns.

Two different disparity maps were used in these experiments. The first generated the familiar "floating square" random-dot stereogram in which the disparity is constant over the entire picture, except for a square in the center of the picture, over which the disparity is a different constant. Such a stereogram is constructed by "sliding" the pixels of square by the disparity and filling in the resulting empty space with new random values. The empty space, which is the result of occlusion, is a place where the disparity is not defined. One question of interest will be what the algorithm does at such locations. This random dot stereogram is shown in Figure 6-1. The second disparity map was computed by smoothing a field of random numbers, resulting in gently rolling random hills. This stereogram is shown in Figure 6-2. The disparity map that generated it is the upper left hand picture of Figure 6-7.

The natural data consisted of aerial views of downtown Washington, D.C. Hand-selected points were used to determine the relative camera parameters, using a program written by Gennery (1980). The images were then rectified by resampling so that they appeared to have been taken by cameras that were parallel. This allowed a simple disparity

map to be used rather than a depth map, as discussed in the previous chapter. Two such stereograms were used; these are shown in Figures 6-3 and 6-4.

In each case, bandpass-filtered images were used in a coarse-fine approach. Bandpass-filtering was chosen over lowpass-filtering (smoothing) because it tends to eliminate any shift in pixel values between the images. For example, edges show up in both images with the same pixel intensity value, namely zero; compare Marr & Poggio's zero-crossing edge detector (1979). Each bandpass-filtered image is computed as follows: the original image is smoothed some number of times (two or three) with a uniform $B \times B$ square window, using the fast constant-time algorithm described in Appendix B. As discussed in that appendix, this is equivalent to convolving the image with a good approximation to a Gaussian function. Then the image is similarly smoothed with a smaller $S \times S$ window, and the result subtracted from the first smoothed image. Since each of the smoothed images have been low-pass filtered with different cutoff frequencies, the difference is a bandpass filtered image, using an approximation to the "difference-of-Gaussians" filter. In each case, $B = 2^k + 1$ and $S = 2^{k-1} + 1$, so that the bandpass region is approximately one octave, and the different resolutions (for different values of k) are separated by about an octave. Such bandpass-filtered images are shown in Figures 6-6 and 6-9. In both figures the image on the left was filtered with $B = 17$ and $S = 9$, while the image on the right was filtered with $B = 33$ and $S = 17$; thus the images represent bands separated by approximately one octave.

Finally, the derivative was estimated using a simple difference in the x direction. Only the x derivative is needed because the images have been rectified so that all disparities are strictly horizontal.

6.2. Synthetic scenes

This section discusses experimental results on the synthetic random dot stereograms. Two such stereograms were used: the random-dot square and the random-dot hills.

Random-dot square. Results obtained by using the algorithm described above on Figure 6-1 are shown in Figure 6-5. This disparity map was computed using a multi-resolution approach. That is, a coarse-resolution disparity map was computed from bandpass-filtered images, and this disparity map was used as input to the next higher frequency range. A uniform disparity map was used as input to the first stage. Successive steps were taken at approximately one octave intervals, using the bandpass-filtered images described above.

Figure 6-5 shows both the computed disparity map and the computed reliability map. Several points should be noted. First, the disparity has been computed essentially perfectly both inside and outside the square. However, the disparity has been incorrectly computed around the boundary of the square, and especially at the right edge where the disparity is undefined due to occlusion. The border inconsistency is due to the fact that there is not enough information from neighboring pixels, due to the disparity discontinuity, to

. disambiguate the highly ambiguous matchings of random dots. The reliability map flags both the border and the occluded regions as unreliable matches. Interestingly, a human observer looking at such a stereogram will see a similarly "crinkled" edge.

Random-dot Hills. The "Random Hills" stereogram is shown in Figure 6-2. The correct disparity map is shown at the upper left of Figure 6-7. The hills portrayed are steep, in that the disparity varies from -16 to $+16$ pixels, and the disparity gradient is as large as 0.9 pixels per pixel. When viewed by a human observer, the hills in this stereogram appear very steep.

Figure 6-7 also shows the result of the algorithm applied to this stereogram under several slightly different conditions. This was in an attempt to evaluate the various parameters associated with the algorithm, for example how many iterations to do at each resolution, what size summing window to use at each step relative to the size of the bandpass filter window, and so on. In each case, the weighting function used was that of equation (2-17) on page 20. The result at the upper right was obtained by the simplest algorithm, in which bandpass-filtered images were used at each step and only one iteration of the algorithm was done at each resolution. The result at the lower left was obtained by taking an initial step using a smoothed image, and using bandpass-filtered images at each subsequent step; moreover, the algorithm was iterated three times at each resolution, using slightly different sized summing windows at each iteration. The result at the lower right also took an initial step using a smoothed image, but only iterated once on each subsequent bandpass-filtered image.

The results presented in this figure span the range of quality, as determined by the root mean squared error between the real disparity and the calculated disparity: from top to left, bottom to right: 0.0, 1.2, 1.9, and 2.7 pixels. As can be seen, even with the correct answer in view, it is difficult to judge subjectively the relative quality of the results. To further show the difference, a scatter plot of calculated disparity against actual disparity for each of the results is shown in Figure 6-7. These plots are shown in Figure 6-8. The upper left plot is in fact the "identity scatter plot," corresponding to the upper left picture of 6-2. The differences between the remaining scatter plots are relatively small, although not negligible. The conclusion is that the method is robust with respect to the parameters mentioned above, as demonstrated by the root-mean-square errors and by the scatter plots, but that some improvement can be obtained by fine-tuning the parameters.

6.3. Washington D.C scenes

This section discusses experiments with the aerial Washington D.C. scenes, shown in Figures 6-3 and 6-4. These stereograms will be referred to as the "Big Washington D.C." stereogram and the "Small Washington D.C." stereogram.

Big Washington D.C. stereogram. As with the other stereograms, bandpass-filtered images were used. Two such images are shown in Figure 6-9. Results from four experiments are shown in Figures 6-10 and 6-11.

In Experiment 1 (Figure 6-10), a relatively simple procedure was used. One iteration of the algorithm was taken at each resolution, with a window size proportional to the bandpass filter window size used to obtain that resolution. For example, at the first step a bandpass-filtered image was constructed as described above with $B = 33$ and $S = 17$; this is the image on the right in Figure 6-9. The algorithm as described by equations (5-8) on page 111 was run once with a summing window 33 pixels wide by 17 pixels high. This procedure was repeated three more times, cutting in half both the smoothing and summing windows each time, yielding the result shown in the figure. (Actually, only half the change in β and γ calculated by equation (5-8) was used.) This disparity map shows the buildings and the streets between them at the top of the picture, but it seems much "rougher" than the image appears.

In Experiment 2 (Figure 6-10), the summing window used at each step was twice as large while the same images were used, so that for example the summing window used at the first step was 65 pixels wide by 33 pixels high, twice as large as in Experiment 1. This can be seen to yield a smoother disparity map, as expected. This disparity map seems better, although there is no objective way to judge.

Experiment 3 (Figure 6-11) produced the best results. This experiment was a sort of hybrid between Experiments 1 and 2, in that at each resolution two iterations were taken, the first with the larger summing window of Experiment 2 and the second with the smaller window of Experiment 1. Moreover, it was determined that the γ (gain) values being calculated were "wild," so only the β (bias) values were calculated while γ was held constant (with $\gamma = 1$). Three steps of this experiment are shown in the figure. Interestingly, the final step is actually worse than the preceding two! This is presumably because a too small summing window is being used at this step. Note that the algorithm has quite clearly picked out the courtyards in the building at the bottom left.

Finally, Experiment 4 (Figure 6-11) was an attempt to use hand-selected initial values for the disparity map. A number of matching points were selected by hand, and at each step the calculated disparity was averaged with a smoothed version of the hand-selected disparity. The procedure followed was like that in Experiment 2. The result of Experiment 4 is very little different than that of Experiment 2, suggesting either that hand-selected initial values are of little help or that they were not incorporated into the method in the right way.

This stereogram has shown that the exact formulation of the method used seems to be more important than it was in the case of the "Random Hills" stereogram. This is presumably because the natural scene presents a more difficult problem, because of a greater amount of detail, some occlusion, and more likelihood that the intensity values for matching features not match. Nevertheless, some promising results have been obtained.

Small Washington D.C. stereogram. The "Small Washington D.C." stereogram (Figure 6-4) was used in a further attempt to incorporate external depth information into the

procedure. In this case, the initial depth values were provided by accurate edge-matchings derived from junction-matchings obtained by the 3D MOSAIC image understanding system (Herman & Kanade, 1984). In particular, the edge matchings provided a number of lines of disparity values in the disparity map; a smoothing operation provided disparity values for points near the lines. The edge matchings all consisted of edges of the buildings near the top of the image.

Figure 6–12 shows two results; the procedures used were identical in the two cases, with the exception that in the left image, initial disparity values provided by Herman were used, and in the right value a uniform field of initial disparity values was used. The results are nearly identical; in each case, the buildings and the streets were picked out. Without accurate ground truth data an objective comparison is impossible. Nevertheless, this result taken together with Experiment 4 of the previous section suggest that externally supplied initial values do not improve the results.

6.4. Summary

This chapter explored by experiment the stereo algorithm developed in the previous chapter. This section summarizes the contributions of this chapter.

It was shown that the method of differences produces excellent results on the "Floating Square" random-dot stereogram. A reliability map was produced in this case that accurately showed the uncertainty in the match at disparity jumps and at occlusions. Good results were also obtained on the "Random Hills" stereogram. This stereogram proved to be relatively insensitive to minor variations in the exact details of the procedure. Promising results on the real data were obtained, although these are difficult to evaluate because of the lack of ground truth data. The real stereograms seemed more sensitive to the exact procedure used; in particular, solving for a field of gain (γ) values seemed to be detrimental. Finally, attempts to improve the results by using independently obtained matches to provide initial disparity estimates didn't seem to help.
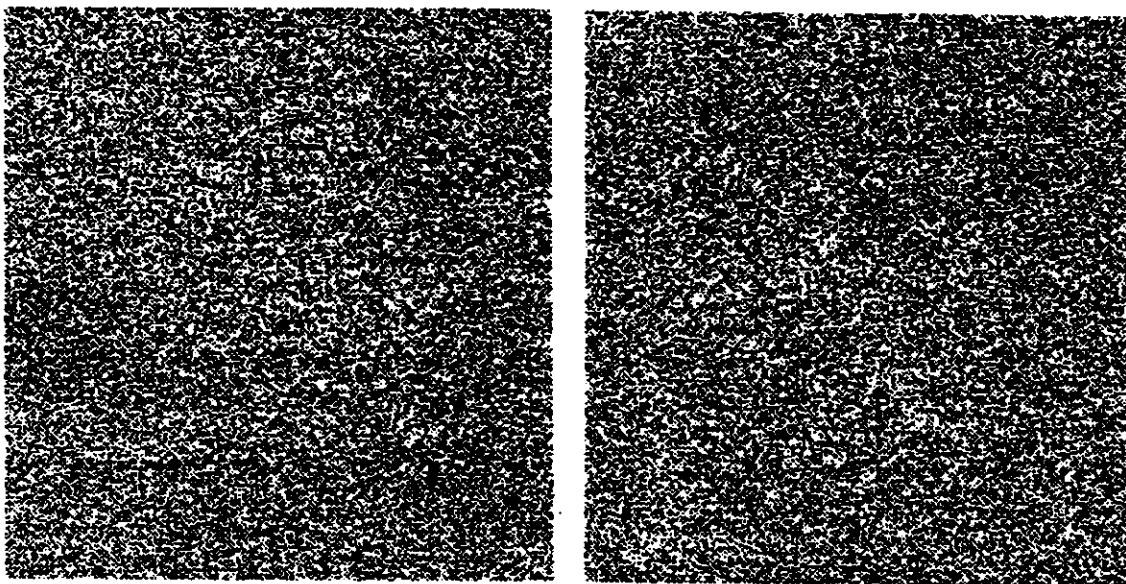
Figure 6–1. "Floating square" random dot stereogram.
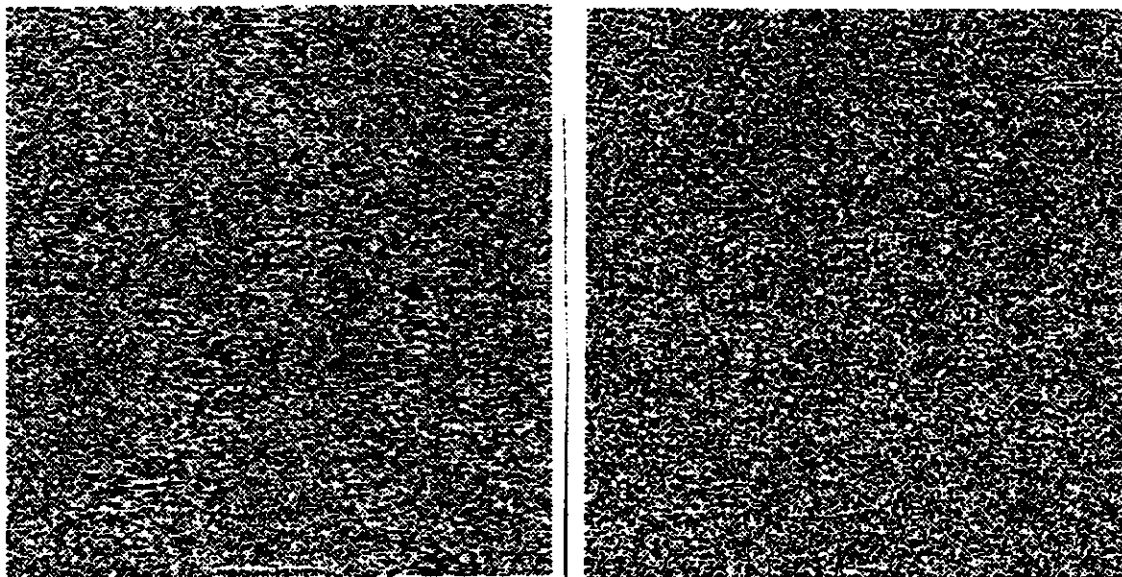
Figure 6–2. "Random Hills" random dot stereogram.

Figure 6-3. "Big Washington D.C." stereogram.
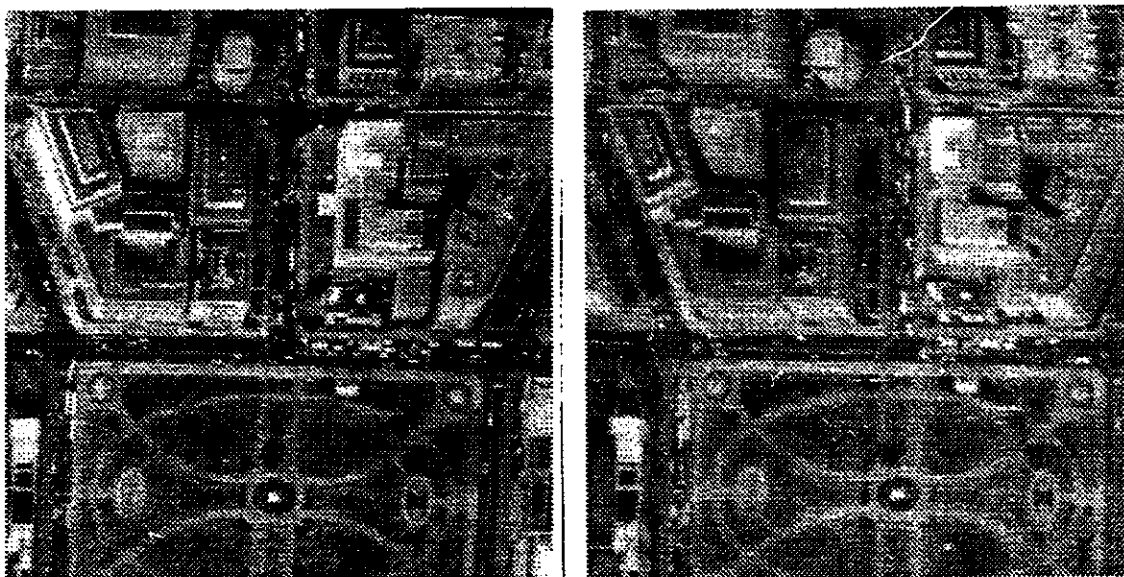
Figure 6-4. "Small Washington D.C." stereogram.
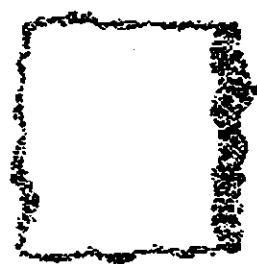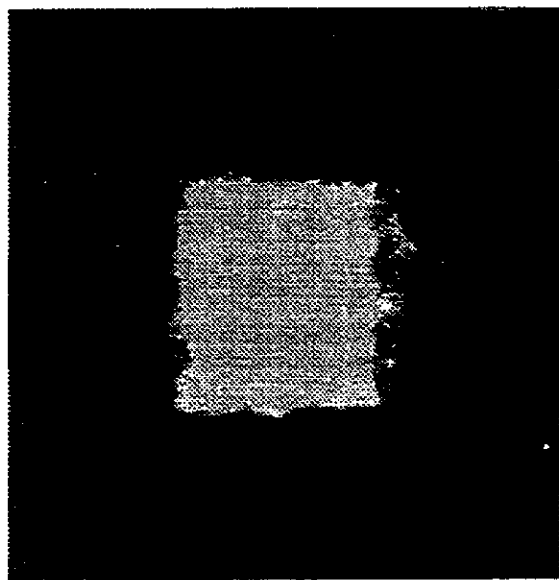
Figure 6–5. Results on "Floating Square" stereogram. Left image is disparity map, right image is reliability map.
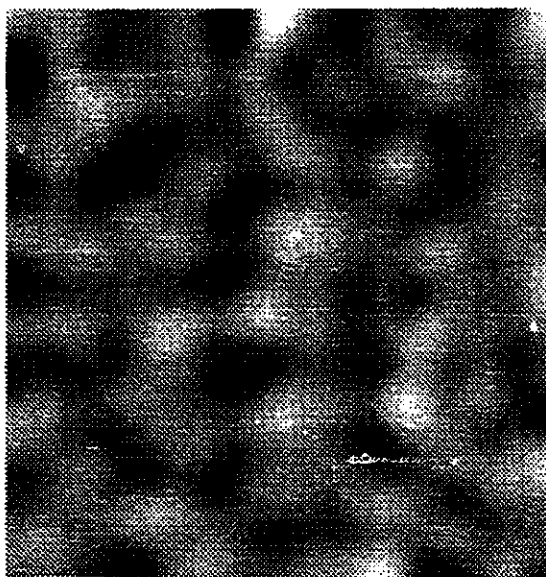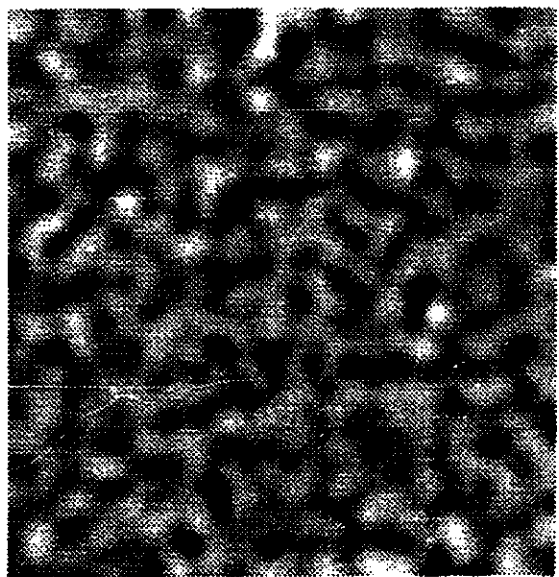
Figure 6–6. Bandpass-filtered version of "Random Hills" stereogram at two frequency ranges. (See text).
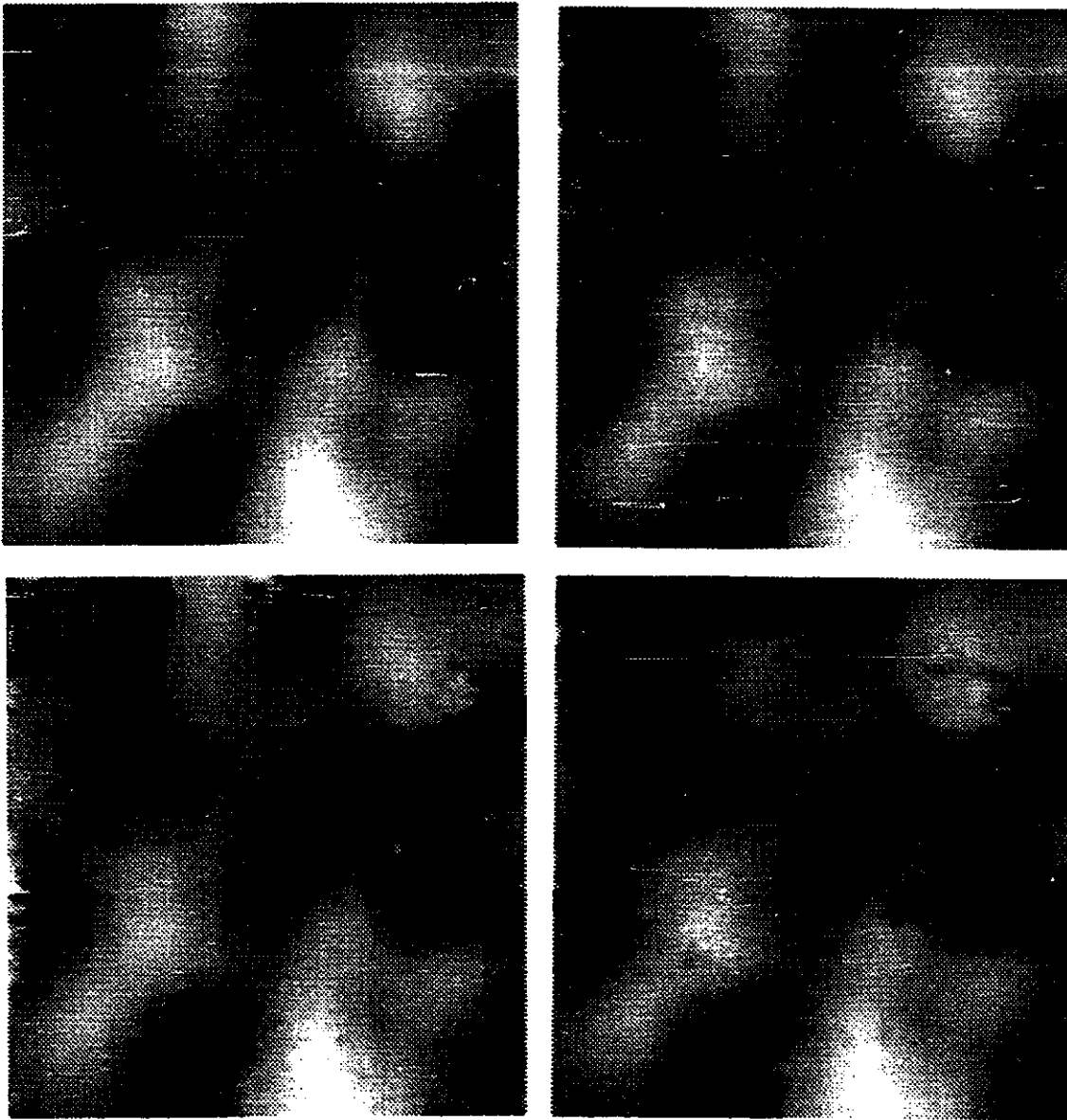
Figure 6-7. Upper left is correct disparity map for "Random Hills" stereogram; the other three are results computed by the method of differences. as discussed in the text.
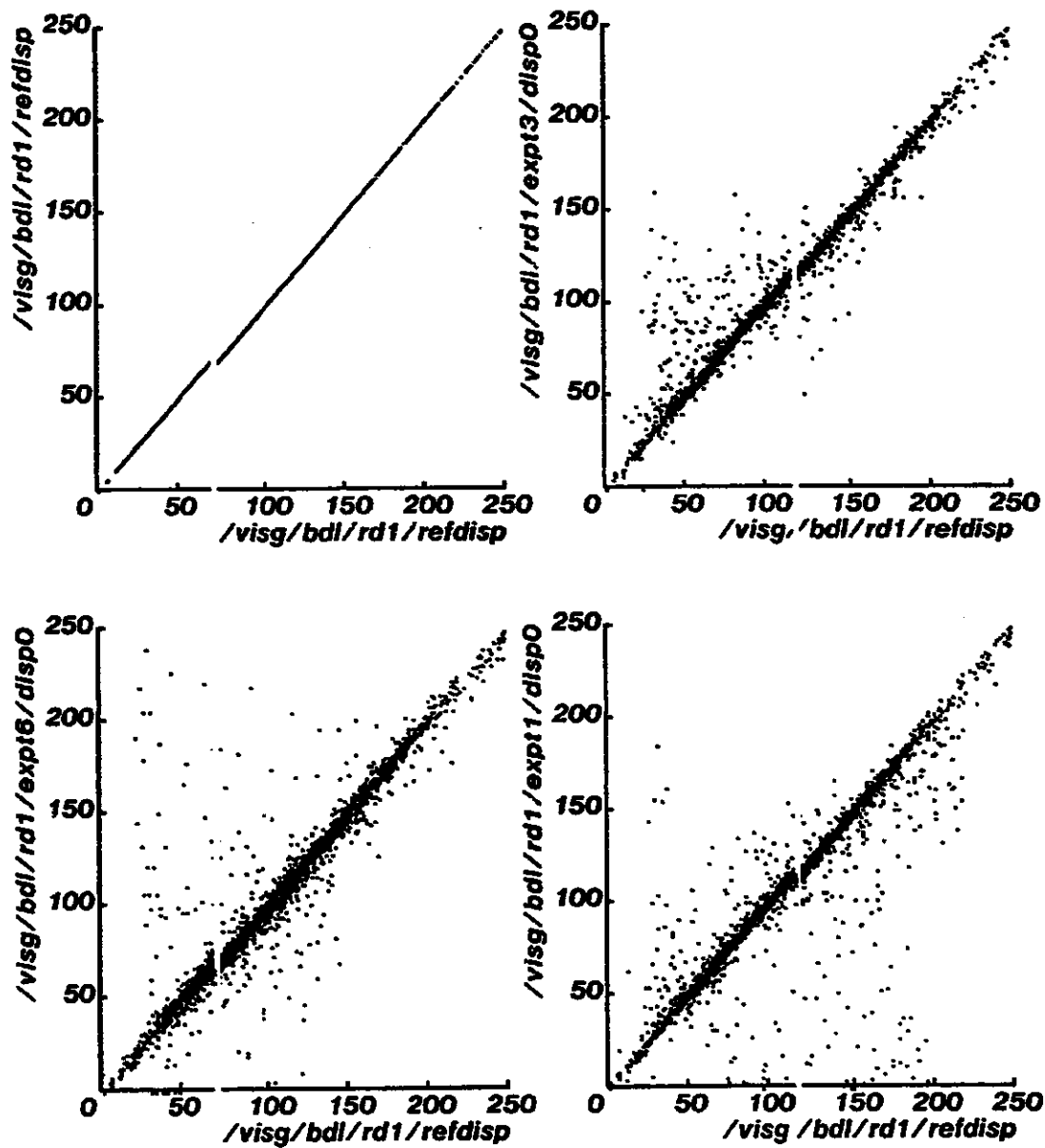
Figure 6–8 .   Scatter plots for each of the disparity maps in Figure 6–7. In each case, the horizontal axis is the actual disparity, and the v ertical axis is the calculated disparity. Unit is $\frac{1}{8}$ pixel. Upper left plot is "identity " scatter plot.

Figure 6–9. Bandpass-filtered version of "Big Washington D.C." stereogram at two frequency ranges. (See text.)
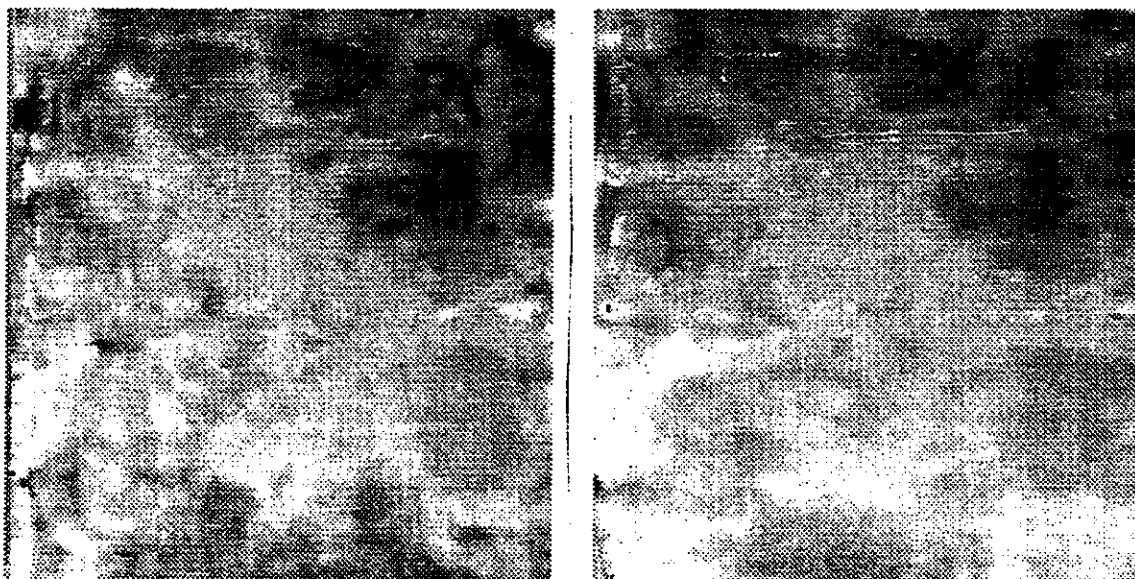
Figure 6–10. Results on "Big Washington D.C." stereogram: Experiment 1 (left) and Experiment 2 (right).
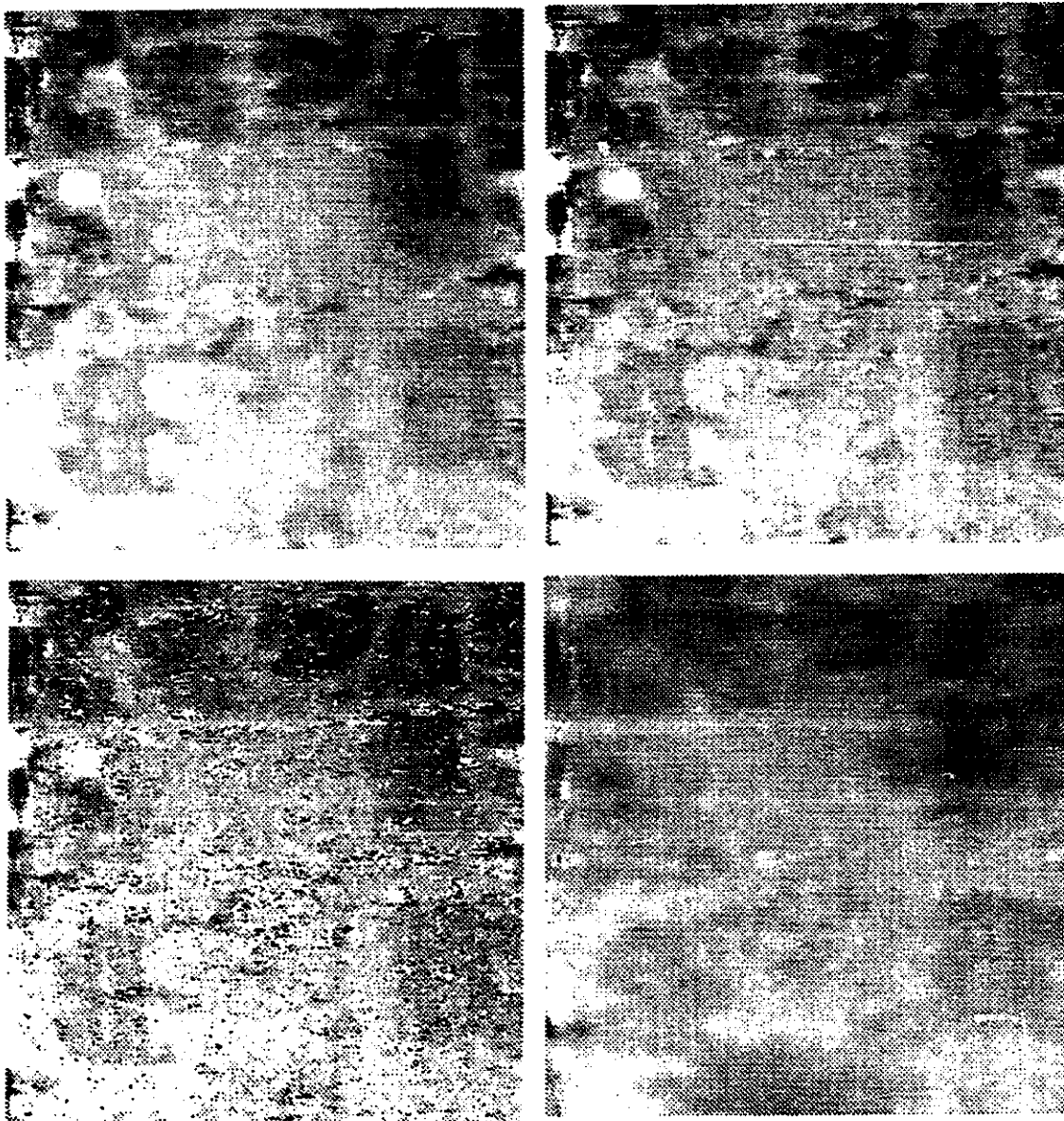
Figure 6–11. More results on "Big Washington D.C." stereogram: three stages of Experiment 3 (top and lower left), and Experiment 4 (lower right).
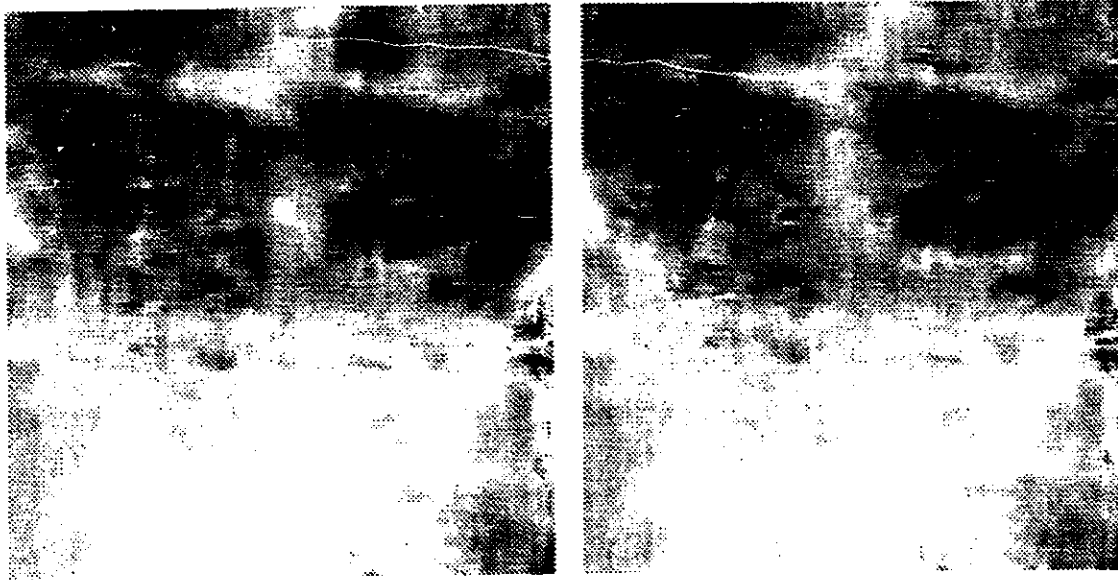
Figure 6–12. Two results on "Small Washington D.C." stereogram. Left disparity used externally generated initial values; right disparity map used uniform initial values.

Chapter 7

Summary and Conclusion

7.1. Introduction

This thesis set out to demonstrate that the method of differences is a useful technique for image registration. This was demonstrated by applying it to two important image matching tasks, namely optical navigation and stereo interpretation. This chapter summarizes what has been shown, discusses its implications, and suggests directions for future research.

7.2. Thesis summary

The first chapter laid the groundwork by defining the traditional image registration problem and characterizing it as a parameter estimation problem, where the parameters are the x and y offsets of the matching parts of the two images. Such a characterization opens the door to generalization. For example, we could characterize the transformation between the images as an affine transformation of coordinates rather than a simple translation. But more importantly, the optical navigation and the stereo interpretation problems can be cast as parameter estimation problems. Thus, they can be solved directly by the method of differences without first solving an intermediate matching problem.

The theory of image matching by the method of differences in a general context was developed in Chapter 2. There we found that each reference point provides one linear constraint on the parameters being solved for. This allows for direct solution of the parameters only where the number of reference points is exactly equal to the number of parameters. In general one will want to use as many reference points as possible to reduce the effect of noise and error. This leads naturally to a least-squares approach. The constraint on the parameters provided by each reference point allows us to estimate the total error resulting from any given change; this estimated error can be minimized by differentiating and setting equal to zero. In the second part of the chapter, a theoretical analysis gave support to the intuitive ideas that smoothing and iteration improve the performance of the algorithm. In particular, the theory shows that behavior of the algorithm depends only on power spectrum of the image, and is independent of phase spectrum. This has two implications. First,

127

the power spectra of images tend to be similar from image to image: they fall off with an approximately exponential decrease as frequency increases, characterized by one or two parameters. This means that the results obtained with the experimental images will be indicative of results obtained with other similar images. Second, it allows general predictions about how smoothing will affect the performance of the algorithm; the prediction is that it will increase the range of convergence. In addition the theoretical analysis provides a prediction of the range and speed of convergence.

In Chapter 3, the theory specifically related to optical navigation by the method of differences was developed. Application of the method requires calculating a matrix representing a system of linear equations in the camera parameters. The computation of this matrix is by far the most time-consuming part of the algorithm, and would be the place where special-purpose hardware could best be put to use. The other main issue to be resolved here was the conditions under which the resulting linear system of equations (in the camera parameters) were numerically stable. The first step was to separate the geometric from the photometric effects: the intuition is that there are certain geometric conditions on the points under which it should be impossible to determine the camera parameters. A measure of geometric stability was developed that satisfied our intuition in two ways: first, it is closely related to the stability of the actual matrix to be inverted, and indeed experiment shows it is a reasonable predictor of that stability. Second, it can be interpreted as a measure of the correlation of the geometric behavior of the parameters being solved for. If two geometric parameters are highly correlated, it should be difficult to solve for them. Analysis of this measure leads to the conclusion that the equations will be numerically unstable when the reference points have a "flat" distribution in three-space. This can occur because the points are in fact on a flat surface, or are well-distributed in three-space but the scene is viewed from a distance with a long-focal length lens. This geometric criterion for the numerical stability of the problem is applicable not only to the method of differences but to any match-based navigation technique.

With this theoretical foundation, Chapter 4 set out to verify the theory by experiment. These experiments were conducted on both real and synthetic data. The main results of these experiments are as follows. First, it was verified that the geometric correlation measure was a good predictor of the the stability of the actual equations. Its advantage of course is that it is independent of the actual scene. The experiments were in agreement with our prediction that "flat" point distributions or scenes viewed from far away resulted in ill-conditioned equations. The adequacy of the condition measure depends on the expected effect of noise on the equations; experiments in which random noise was added to the pictures shows that the observed conditioning was satisfactory. The range of convergence for one-parameter estimation was determined to be as follows: for pan and tilt, anywhere from ± 10 degrees to ± 50 degrees, depending on the scene; for roll, up to $\pm$ 30 degrees independent of the scene. For the position parameters (x, y, and z) the tolerance was as much as ± 1 meter in a room-sized scene. These ranges of course depend on the degree of smoothing; the amounts mentioned here correspond to very large smoothing windows, as much as $\frac{1}{4}$ the image size. A wider angle view allows larger smoothing windows and

thus more tolerance, but decreases accuracy for a given retina. In the multi-parameter case the range of convergence became smaller as the number of parameters was increased, but retained a useful range even in the six-parameter case. The accuracy observed was essentially that expected for one-pixel accuracy in the individual matches. For typical images, accuracy was about ±1 or 2 cm in x and y. Less accuracy was obtained in z, which suggests the possibility of side-looking cameras to obtain accuracy in this direction. Angular accuracy was extremely high. This accuracy is due in part to using reference points near edges that are unaffected by photometric error. Finally, the method involves accumulating a few dozen quantities per reference point per iteration, and so is quite fast. These properties also make it suitable for future implementation on special-purpose hardware. This does not include the time required to smooth the images, but hardware techniques for doing so are well-understood and available in commercial units.

Chapter 5 discussed the version of the algorithm to be used for computing a stereo disparity map. While a navigation algorithm seeks to estimate a few global parameters, a stereo algorithm computes a field of local parameters; each parameter is the local displacement of the image due to binocular disparity, and is a direct measure of the distance to the point. The approach taken is to use the method of differences to match a small image patch around each given point against the other image. This process is repeated iteratively, yielding better and better disparity estimates. Actually, the match is at each iteration is done against the other image as distorted by the current disparity field estimate. As the process is iterated, successively less smooth images and smaller matching windows are used. This allows correct matching to take place even though the images may distorted relative to each other due to perspective. An algorithm based on an efficient technique for image smoothing (described in Appendix B) allows this to be done efficiently and in time independent of the window size. The method requires about 30 sec per iteration for a 250×250 image on a VAX 11/780 (independent of smoothing time). Its regular structure makes it quite suitable for implementation on special-purpose hardware.

Finally, Chapter 6 presents the results of experiments designed to test the stereo algorithm. Again, both synthetic and real data were used. The synthetic data consisted of random-dot stereograms, of both the "floating-square" type and of the "rolling-hills" type. Excellent disparity maps were obtained in both cases. The performance of the algorithm seems to be fairly robust with respect to the details of the algorithm, such as the size of smoothing window used. The real data consisted of aerial views of Washington, D.C. Promising results were obtained: buildings and streets were detected. An experiment to determine whether matches provided by an independent source (another stereo matching program) could improve the performance of the method-of-differences algorithm were inconclusive. The problem with evaluating the results on the real images was the lack of ground-truth data against which to judge them, or of a vision system to take disparity maps as input for some image understanding task. Nevertheless, the results seem promising.

7.3. Conclusions

This thesis has investigated the usefulness of the method of differences as an image matching technique. In particular the method has been applied to two problems: optical navigation and stereo vision. We have seen that the technique is applicable in any situation where a (very) rough estimate of the match is available, but an accurate answer is desired. A substantial part of the research has been directed toward determining how rough the estimate may be and how accurate the final answer is. It has been demonstrated that that the method has adequate range and accuracy for many robotic tasks, particularly as applied to optical navigation.

This research has revealed several factors that are essential to making the method of differences work. We have seen the importance of smoothing and of iteration to the method. Roughly speaking, without smoothing the technique has too little range and without iteration it has too little accuracy to be useful in most applications. Because smoothing reduces the accuracy of the method, different degrees of smoothing can be used at each iteration, to yield a coarse-fine method. Moreover, the thesis has recognized the the importance of using points near intensity edges because of the relative unambiguity of matches for such points. This fact has long been recognized by advocates of edge-based processing techniques. However, as this research has shown, the importance of points near edges does not demand edge-based algorithms.

The method provides several advantages over other matching techniques. It is free of search, which can be impractical because of its expense in multi-dimensional parameter spaces. With a standard search, the expense goes up like the size of the volume to be searched, which is as the power of the dimensionality of the parameter space; whereas the expense of the method of differences is roughly a function only of the distance of the initial estimate from the actual answer. Thus, the method of differences has its greatest advantage in high-dimensional parameter problems, such as navigation. However, it provides an advantage even in low-dimensional spaces, such as the disparity in a stereo depth map. This is because the structure of the algorithm allows for an efficient implementation, as we saw in the computation of the depth map. Furthermore, for parameter estimation problems it computes exactly what is needed without proceeding through intermediate results, such as point matches or an optical flow field. Finally, its regular and simple structure make it quite suitable for implementation on special-purpose hardware, as we will see in the next section.

These advantages are balanced by a few restrictions on the use of the method of differences. The primary restriction is that an initial estimate of the match parameters must be available. However, in the optical navigation application this restriction is generally not a problem; it also did not seem to present difficulties in the stereo case. Another restriction is that the navigation application requires reference points with known z values. In the manufacturing scenario, these can be obtained as part of a training step.

7.4. Future research

The research described here can be carried forward in four ways: the algorithms themselves can be improved, our theoretical understanding of them can be improved, they can be implemented on special-purpose hardware, and the method can be tried on new applications.

There is still room for improvement in the algorithms themselves. Chapter 2 suggested weighting the contributions of the points, but this was not actually carried out in the navigation case. Experiments are in order to determine whether weighting helps and what the best function to use is, and whether using the photometric bias and gain parameters would improve the results. Other ideas such as use of multiple cameras and decoupling the solution of the parameters have been suggested but not tried. Much tuning of the stereo algorithms is needed. For example, the proper relationship between the iteration number and the sizes of smoothing window and of the matching window is not understood; this could be determined by experiment. Also, the experiment that used externally provided matches for the initial guess suggested that they did not help; this is counter-intuitive, and may be due to the stereo algorithm's not being tuned properly.

The theory discussed here remains to be improved in several ways, all revolving around the analysis of the method in terms of the power spectrum given in equation (2–44). This equation could be used to make more precise predictions than the general ones made in Chapter 2, if the theory could be developed further. Furthermore, a similar equation that would take weights into account would advance our understanding of whether the weights help. Finally, an understanding of the relationship between the power spectrum of the disparity map and the performance of the stereo algorithm might be forthcoming from a more advanced form of the equation.

The third area of research was suggested above when it was mentioned that the algorithm is suitable for implementation in special-purpose hardware. This remains to be shown by doing such an implementation. Two types of architectures could be envisioned: in a picture-parallel design, each component is capable of performing all the operations of the algorithm on its portion of the picture; in an operation-parallel design, each component does some subset of the operations on all of the data, and passes its results on to other components. In either case, an algorithm that is free of decisions in the inner loop is especially suitable: in the first case because this allows the use of a single instruction stream, and in the second case because decisions can interrupt the flow of data in the pipeline and reduce throughput. A sufficiently fast hardware design raises the fascinating possibility of a real-time optical feedback for a robotic device, such as an arm. By reparameterizing the algorithm, it could be modified to directly calculate the joint rotation parameters instead of the six position parameters. The acid test here would be to move an object around in three space while a robot arm equipped with a camera follows it in real time.

Finally, as was mentioned in the introduction there are many computer vision problems besides the two considered in this thesis in which matching plays an important role. These include object identification, motion understanding, and many others. Any problem which can be characterized as one of finding parameters of transformation between two images

is a candidate for the method of differences. Some problems which have not traditionally been treated as matching problems can in fact be cast as such. For example, finding axes of symmetry in an image is essentially a self-matching problem; cast in the more general terms of finding parameters of transformations of symmetry, this suggests that the method of differences could be fruitfully applied to very general symmetry transformation problems. The missing element is a method for finding a reasonable approximation to the symmetry transformation.

This thesis set forth to demonstrate the value of the method of differences as a matching technique. If the theory and experiments described here provide the inspiration for others to apply the method to their own vision problems, then it will have succeeded in its goal.

Appendix A

Notation

Vector notation. Vectors are indicated by bold-face letters, such as **p**. Two- and three-space points are represented by row vectors with subscript x, y, and z used to select components, for example

$$\mathbf{p} = [\mathbf{p}_x \quad \mathbf{p}_y \quad \mathbf{p}_z].$$

Numerical subscripts are used to select components of vectors in other cases. Row vectors, while contrary to customary usage, allow the differentiation operator described below to be a prefix operator.

The differentiation operator D and the chain rule. The familiar notions of a derivative of a scalar with respect to another scalar and the chain rule can be extended to the "derivative" of a vector-valued function with respect to a vector-valued argument and a corresponding chain rule. Consider the vectors

$$\mathbf{x} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_l],$$
$$\mathbf{y} = \mathbf{y}(\mathbf{x}) = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}_m], \quad \text{and}$$
$$\mathbf{z} = \mathbf{z}(\mathbf{y}) = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_n].$$

These are row vectors so that we can define a prefix operator $D_\mathbf{x}$ by

$$D_\mathbf{x} = [D_{x_1} \quad D_{x_2} \quad \cdots \quad D_{x_l}]^\mathsf{T}.$$

Here D_t denotes the familiar operator of partial differentiation $\partial/\partial t$ with respect to the scalar t. Note that while $\mathbf{x}$ is a row vector, $D_\mathbf{x}$ is a column vector; think of the x in $D_\mathbf{x}$ as being in the denominator like the t in $D_t = \partial/\partial t$. $D_\mathbf{x}$ is also known as the gradient operator with respect to $\mathbf{x}$. By defining it in this way we can write

$$D_\mathbf{x}\mathbf{y} = \begin{bmatrix} D_{x_1} \\ D_{x_2} \\ \vdots \\ D_{x_l} \end{bmatrix} [y_1 \quad y_2 \quad \cdots \quad y_m] = \begin{bmatrix} D_{x_1}y_1 & D_{x_1}y_2 & \cdots & D_{x_1}y_m \\ D_{x_2}y_1 & D_{x_2}y_2 & \cdots & D_{x_2}y_m \\ \vdots & \vdots & \ddots & \vdots \\ D_{x_l}y_1 & D_{x_l}y_2 & \cdots & D_{x_l}y_m \end{bmatrix}.$$

This $l \times m$ matrix is also known as the Jacobian. Its primary use is in the chain rule:

$$D_x z = (D_x y)(D_y z).$$

The left-hand side is an $l \times n$ matrix, and the right-hand side is a product of an $l \times m$ and an $m \times n$ matrix. Moreover, by virtue of the Taylor series, we can write

$$\Delta y \approx \Delta x D_x y,$$

where of course $\Delta y = y(x + \Delta x) - y(x)$.

The rotation matrices. The rotation matrices are written out here in full. They define the sense and meaning of the rotation angles.

$$
A_{PA} = \begin{bmatrix} \cos(\alpha_{PA}) & 0 & \sin(\alpha_{PA}) \\ 0 & 1 & 0 \\ -\sin(\alpha_{PA}) & 0 & \cos(\alpha_{PA}) \end{bmatrix}, \quad
A'_{PA} = \begin{bmatrix} -\sin(\alpha_{PA}) & 0 & \cos(\alpha_{PA}) \\ 0 & 0 & 0 \\ -\cos(\alpha_{PA}) & 0 & -\sin(\alpha_{PA}) \end{bmatrix}.
$$

$$
A_{TI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_{TI}) & +\sin(\alpha_{TI}) \\ 0 & -\sin(\alpha_{TI}) & \cos(\alpha_{TI}) \end{bmatrix}, \quad
A'_{TI} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin(\alpha_{TI}) & +\cos(\alpha_{TI}) \\ 0 & -\cos(\alpha_{TI}) & -\sin(\alpha_{TI}) \end{bmatrix}.
$$

$$
A_{RO} = \begin{bmatrix} \cos(\alpha_{RO}) & \sin(\alpha_{RO}) & 0 \\ -\sin(\alpha_{RO}) & \cos(\alpha_{RO}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad
A'_{RO} = \begin{bmatrix} -\sin(\alpha_{RO}) & \cos(\alpha_{RO}) & 0 \\ -\cos(\alpha_{RO}) & -\sin(\alpha_{RO}) & 0 \\ 0 & 0 & 0 \end{bmatrix}.
$$

$$
A_{AZ} = \begin{bmatrix} \cos(\alpha_{AZ}) & 0 & -\sin(\alpha_{AZ}) \\ 0 & 1 & 0 \\ \sin(\alpha_{AZ}) & 0 & \cos(\alpha_{AZ}) \end{bmatrix}, \quad
A'_{AZ} = \begin{bmatrix} -\sin(\alpha_{AZ}) & 0 & -\cos(\alpha_{AZ}) \\ 0 & 0 & 0 \\ \cos(\alpha_{AZ}) & 0 & -\sin(\alpha_{AZ}) \end{bmatrix}.
$$

$$
A_{EL} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_{EL}) & -\sin(\alpha_{EL}) \\ 0 & \sin(\alpha_{EL}) & \cos(\alpha_{EL}) \end{bmatrix}, \quad
A'_{EL} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin(\alpha_{EL}) & -\cos(\alpha_{EL}) \\ 0 & \cos(\alpha_{EL}) & -\sin(\alpha_{EL}) \end{bmatrix}.
$$

Appendix B

Smoothing the Image and Computing the Gradient

The techniques discussed in this thesis require that the images be smoothed, and that the image gradient be computed. This appendix shows how these operations can be accomplished efficiently. The first section discusses image smoothing; the second section presents a method of computing the gradient based on the smoothing technique.

Smoothing the image. The basis of a number of fast smoothing techniques is a technique for uniform smoothing, that is replacing each pixel by the average of the pixels in a square window around it, in time essentially independent of the size of the window. This technique has been described elsewhere (Price, 1976), but it bears repeating here. The following equations actually describe how to compute the sum over a $(2n + 1) \times (2n + 1)$ window centered on each pixel; the average is of course $1/(2n + 1)^2$ times the sum.

First consider the one-dimensional problem: given an array of numbers I_k, compute the local sum:
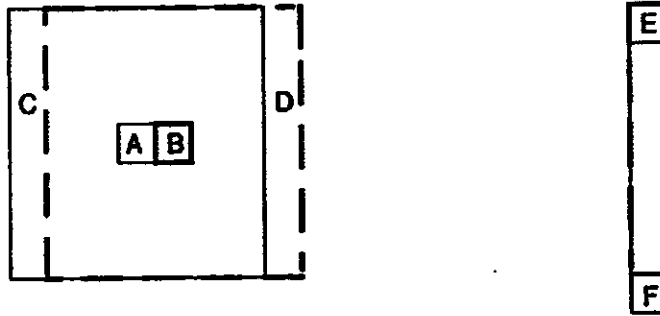
$$S_k = \sum_{k'=k-n}^{k+n} I'_k. \tag{B-1}$$

Observe that

$$S_k = S_{k-1} - I_{k-n-1} + I_{k+n}. \tag{B-2}$$

Thus, if we compute S_0 using (B-1), the rest of the S_i can be computed using (B-2) with only one addition and one subtraction per step, regardless of the size of n.

In two dimensions, we wish to compute

$$S_{k,l} = \sum_{l'=l-n}^{l+n} \sum_{k'=k-n}^{k+n} I_{k',l'}.$$

135

Figure B–1. Two additions and two subtractions per pixel suffice to compute the sum of pixels in a square window around each pixel in the image, independent of size of window. Left: solid lines indicate area to be summed around pixel A; dashed lines indicate area to be summed around pixel B. Sum for B can be obtained from sum for A by subtracting "column sum" indicated by C and adding column sum indicated by D. An array of column sums is maintained, one for each column, centered about the current row. Right: column sum needed for current row (solid line) is updated to column sum for next row (dashed line) by subtracting pixel E and adding pixel F.

This can be written as

$$S_{k,l} = \sum_{l'=l-n}^{l+n} R_{k,l'}, \quad \text{where}$$

$$R_{k,l} = \sum_{k'=k-n}^{k+n} I_{k',l}.$$

Then

$$R_{k,l} = R_{k-1,l} - I_{k-n-1,l} + I_{k+n,l}, \quad \text{and}$$

$$S_{k,l} = S_{k,l-1} - R_{k,l-n-1} + R_{k,l+n}.$$

Thus, the two-dimensional sum can be computed with only two additions and two subtractions per pixel, independent of n. Temporary storage of one row of elements is required to hold $R_{k,l}$ for a given k while the $S_{k,l}$ for that k are being computed. The operation of this algorithm is illustrated in Figure B–1.

The uniform average is useful in itself, but is also useful for computing an approximation to Gaussian smoothing. This is accomplished by taking advantage of the central limit theorem: a function convolved with itself N times approaches a Gaussian curve as N goes to infinity. Thus, convolution with a Gaussian can be accomplished by repeated convolution
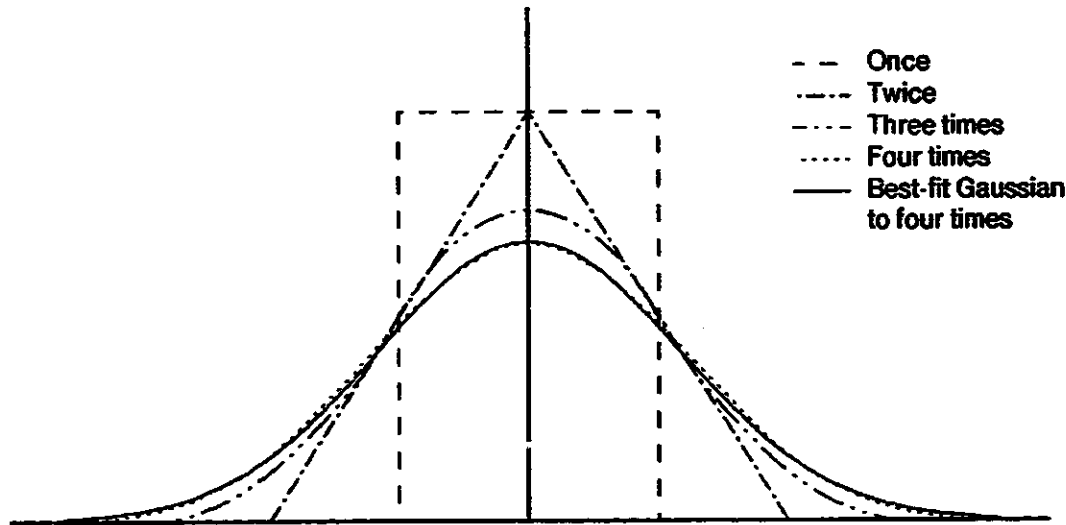
Figure B–2. Averaging with uniform mask once, twice, three and four times is equivalent to convolving with functions shown. These functions converge to a Gaussian. The best-fit Gaussian to the fourth case is shown.

with a uniform window, that is, by repeated uniform averaging. For practical purposes, convolution with a uniform window three or four times suffices. The result of convolving a uniform window with itself, one, two, three, and four times is compared with a Gaussian in Figure B–2.

Computing the image gradient.   In addition to smoothing, the algorithms discussed in this thesis require a technique for computing the image gradient. Computing the gradient is typically done by looking at a few pixels surrounding each pixel in the image. Unfortunately, this only takes account of information near the given pixel, and thus does not necessarily serve as a good estimate of the behavior of the image further away than the support of the gradient function. This problem could be alleviated by taking the gradient of the smoothed images. To avoid round-off error, this must be done at high precision. For example, we can replace the average with the sum and only divide by the size of the window at the very last step. But this technique has pitfalls as well. For example, if we use the uniform averaging window, and compute the gradient by taking the difference between the average at two adjacent pixels, then the contributions of all pixels except those at a distance of n from the given pixel cancel, so the estimate of the derivative at k is $(I_{k+n} - I_{k-n})/(2n)$. But a gradient estimate that depends only on pixels far away from the given pixel doesn't seem right either. One is also left with the problem of interpolating the image value and gradient between pixels for subpixel accuracy.

Instead, this thesis adopts an approach based on an understanding of what purpose

$$p \quad p+e$$
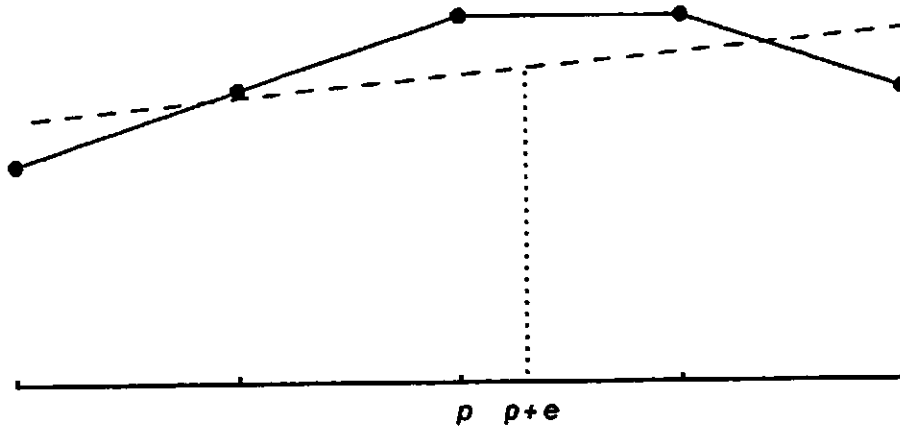
Figure B-3. Smoothing and gradient estimation can be accomplished by fitting a straight line (dashed line) to the intensity values (solid line). To obtain intensity value and gradient at $p+\epsilon$ (dotted line), a line is fitted to points centered on p. Slope is gradient estimate, height at $p + \epsilon$ is interpolated intensity. The text describes how to compute this in time independent of size of neighborhood.

the gradient serves: the gradient at a given point is a first-order estimate of the behavior of the image as the matching point is moved around near the given point. Thus, it would seem natural to fit a plane to the points in the vicinity of the given point, and to use its height as an estimate of the value of the image at the given point, and its gradient as an estimate of the image gradient. Note that this gives us a natural solution to the problem of interpolating the intensity and gradient, since the fitted plane gives an intensity value and gradient at every point, even those between pixels. The size of the neighborhood over which the plane is fitted depends on the size of movement to be accommodated in the estimate of the matching position. See Figure B-3.

Fortunately, the gradient field can be computed in this manner independent of the size of the support for the plane. Suppose we wish to compute the interpolated intensity value and gradient of an image I at a point $(x + \epsilon_x, y + \epsilon_y)$, whose nearest integer pixel is (x, y). Thus, we wish to fit a plane to the square of points (x', y') in a square neighborhood of points centered on the pixel (x, y). Then the x and y components of the gradient of the fitted plane are respectively

$$\hat{I}_x = \frac{\displaystyle\sum_{x'=x-n}^{x+n} \sum_{y'=y-n}^{y+n} (x' - x)I(x',y')}{\displaystyle\sum_{x'=x-n}^{x+n} \sum_{y'=y-n}^{y+n} (x' - x)^2} = \frac{S_x(x,y) - xS(x,y)}{D_x}, \tag{B-3}$$

$$\hat{I}_y = \frac{\displaystyle\sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n}(y'-y)I(x',y')}{\displaystyle\sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n}(y'-y)^2} = \frac{S_y(x,y)-yS(x,y)}{D_y}, \qquad\qquad (B\text{-}4)$$

where

$$S(x,y) = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} I(x',y'),$$

$$S_x(x,y) = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} x'I(x',y'),$$

$$S_y(x,y) = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} y'I(x',y'),$$

$$D = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} 1 = 1/(2n+1)^2,$$

$$D_x = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} (x'-x)^2,$$

$$D_y = \sum_{x'=x-n}^{x+n}\sum_{y'=y-n}^{y+n} (y'-y)^2.$$

Moreover, the estimated (interpolated) value of intensity at the point (x,y) is just the average over the square neighborhood around (x,y), so that the estimated intensity at $I(x+\epsilon_x, y+\epsilon_y)$ is

$$\hat{I} = \frac{S(x,y)}{D} + \epsilon_x\hat{I}_x + \epsilon_y\hat{I}_y. \qquad\qquad (B\text{-}5)$$

Now, the denominators D, D_x, and D_y are constants independent of x and y, and so can be precomputed. Moreover, since the range of (x',y') near (x,y) is the same rectangular neighborhood for each (x,y), the fields of sums $S(x,y)$, $S_x(x,y)$, and $S_y(x,y)$ can be precomputed using the incremental techniques described previously the in time independent of the size of the support. Then, when we need to compute $\hat{I}$, $\hat{I}_x$, and $\hat{I}_y$, we can plug these sums into equations (B-3), (B-4), and (B-5).

References

J. K. Aggarwal, L. S. Davis, and W. N. Martin (1981). Correspondence processes in dynamic scene analysis. *Proceedings of the IEEE* 69 562-572.

H. H. Baker and T. O. Binford (1981). Depth from edge and intensity based stereo. *Proceedings of the 7th International Joint Conference on Artificial Intelligence* 631-636.

D. H. Ballard and O. A. Kimball (1983). Rigid body motion from depth and optical flow. *Computer Vision, Graphics, and Image Processing* 22 95-115.

D. I. Barnea and H. F. Silverman (1972). A class of algorithms for fast digital image registration. *IEEE Transactions on Computers* C21 179-186.

A. R. Bruss and B. K. P. Horn (1983). Passive navigation. *Computer Vision, Graphics, and Image Processing* 21 3-20.

C. Cafforio (1979). *Remarks on the differential method for the estimation of movement in television images.* Presented at Picture Coding Symposium, Ipswich.

C. Cafforio and F. Rocca (1979). Tracking moving objects in television images. *Signal Processing* 1 133-140.

N. H. Cornelius and T. Kanade (1983). Adapting optical-flow to measure object motion in reflectance and x-ray image sequences. *Proceedings of the ACM SIGGRAPH/SIGART Workshop on Motion: Representation and Perception*, Toronto, 50-58.

G. Dahlquist and Å. Björck (1974). *Numerical Methods.* Prentice-Hall.

C. L. Fennema and W. B. Thompson (1979). Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing* **9** 301-315.

D. B. Gennery (1980). Modeling the environment of an exploring vehicle by means of stereo vision. PhD Thesis, Department of Computer Science, Stanford University.

Grimson, W. E. L. (1981). A computational theory of visual surface interpolation. MIT AI Memo 613.

U. V. Helava (1978). Digital correlation in photogrammetric instruments. *Photogrammetria* **34** 19-41.

R. L. Henderson, W. J. Miller, and C. B. Grosch (1979). Automatic stereo reconstruction of man-made targets. *SPIE* **186** 240-248.

M. Herman and T. Kanade (1984). The 3D MOSAIC scene understanding system: incremental reconstruction of 3D scenes from complex images. Technical Report CMU-CS-84-102, Department of Computer Science, Carnegie-Mellon University.

B. K. P. Horn and B. G. Schunck (1981). Determining optical flow. *Artificial Intelligence* **17** 185-203.

C. V. P. Hough (1962). Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

T. S. Huang and R. Y. Tsai (1981). Image sequence analysis: motion estimation in image sequence analysis. In *Image Sequence Analysis*, Springer Series in Information Sciences, Vol. 5, Ch. 1, Springer-Verlag.

B. Julesz (1960). Binocular depth perception of computer-generated patterns. *Bell System Technical Journal* **39** 1125-1161.

B. Julesz (1962). Towards the automation of binocular depth perception. *Proceedings of the IFIP Congress* 439-443.

E. R. Kretzmer (1952). Statistics of television signals. *Bell System Technical Journal* **31** 751-763.

D. N. Lee (1980). The optic flow field: the foundation of vision. *Philosophical Transactions of the Royal Society of London* **B290** 169-179.

J. O. Limb and J. A. Murphy (1975a). Measuring the speed of moving objects from television signals. *IEEE Transactions on Communications*, COM-23 474-478.

J. O. Limb and J. A. Murphy (1975b). Estimating the velocity of moving images in television signals. *Computer Graphics and Image Processing* 4 311-327.

B. D. Lucas and T. Kanade (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence.*

D. Marr and T. Poggio (1979). A computational theory of human stereo vision. *Proceedings of the Royal Society of London* B 204 301-328.

D. Marr and T. Poggio (1976). Cooperative computation of stereo disparity. *Science* 194 283-287.

H. P. Moravec (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Robotics Institute, Carnegie-Mellon University.

K. Mori, M. Kidode, and H. Asada (1973). An iterative prediction and correction method for automatic stereocomparison. *Computer Graphics and Image Processing* 2 393-401.

R. Nevatia (1976). Depth measurement by motion stereo. *Computer Graphics and Image Processing* 5 203-214.

Y. Ohta and T. Kanade (1983). Stereo by intra- and inter-scanline search using dynamic programming. Technical Report CMU-CS-83-162, Computer Science Department, Carnegie-Mellon University.

A. V. Oppenheim and J. S. Lim (1981). The importance of phase in signals. *Proceedings of the IEEE* 69 529-541.

S. W. Pizer (1975). *Numerical Computing and Mathematical Analysis.* Science Research Associates, Chicago.

K. Prazdny (1981). Determining the instantaneous direction of motion from optical flow generated by a curvilinear moving observer. *Computer Graphics and Image Processing* 17 238-248.

K. E. Price (1976). Change detection and analysis in multi-spectral images. PhD Thesis, Department of Computer Science, Carnegie-Mellon University.

D. R. Reddy and S. Rubin (1978). Representation three-dimensional objects. Technical Report CMU-CS-78-113, Department of Computer Science, Carnegie-Mellon University.

W. C. Rheinboldt (1974). Methods for solving systems of nonlinear equations. *Regional Conference Series in Applied Mathematics* 14. Society for Industrial and Applied Mathematics.

D. Terzopoulos (1983). Multi-resolution computation of visible-surface representations. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

R. Y. Tsai and T. S. Huang (1980). Three-dimensional motion estimation. In *Signal Processing: Theories and Applications*. M. Kunt and F. De Coulon (eds.). North Holland. 263-269.

R. Y. Tsai and T. S. Huang (1981). Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces. *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, Las Vegas.

J. Webb and J. K. Aggarwal (1983). Shape and correspondence. *Computer Vision, Graphics, and Image Processing* 21 145-160.

M. Yachida (1983). Determining velocity maps by spatio-temporal neighborhoods from image sequences. *Computer Vision, Graphics, and Image Processing* 21 262-279.

Y. Yakimovsky and R. Cunningham (1978). A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing* 7 195-210.

Thesis Summary

1. Introduction

Image matching (or registration) is, in general terms, aligning two identical or similar images or parts of images. The subject of this thesis is a class of techniques for doing image registration by the *method of differences*. The method of differences is a matching technique that uses the image intensity gradient together with the intensity differences between the images in a procedure that iteratively improves an initial estimate. The thesis will develop a number of algorithms for various kinds of matching based on this method, and show how they can be applied to optical navigation and stereo image interpretation, both in theory and by experiment. This summary discusses the importance of image registration, formally defines it and generalizes that definition, describes the method of differences, and finally summarizes the thesis chapter by chapter.

2. Motivation

Image matching is basic to a number of vision problems. These include optical navigation (also known as motion analysis), stereo image interpretation, object analysis, change detection, and others. The first two, optical navigation and stereo image interpretation, are two of the most important; they are the two applications considered in this thesis.

Optical navigation refers to the guidance of a robot, such as a robot arm or an autonomous roving vehicle, by means of input from an optical sensor such as a camera. Robots need navigational feedback from their environment because the environment is not perfectly predictable, and because the robot's response to a command to move in a certain way is not perfectly predictable. Optical navigation is one of many ways of providing that feedback. The optical domain is particularly rich in information, but therefore correspondingly difficult for computer analysis. It is the aim of the method of differences to provide a relatively inexpensive way to do optical navigation.

The primary problem of optical navigation is this: given two views of the same scene from two different cameras, determine the parameters describing the relationship between the coordinate systems of the cameras. This problem has two essential characteristics: it

is a parameter estimation problem, and in most applications a reasonable estimate of those parameters is available at the outset. As we will see, these two characteristics make this problem particularly suited to solution by the method of differences.

The objective of stereo vision (or more precisely, binocular vision) is to obtain information about the three-dimensional form of an object or a scene from two camera views. Binocular vision is one of many sources of information available about the three-dimensional form of the world. The need for reconstructing this information from camera views arises essentially due to a shortcoming in the sensors used: cameras record only a two-dimensional projection of a three-dimensional world. Thus one method of attacking this problem has been to overcome this deficiency, for example through structured lighting, sonar range detectors, contact sensors, and so on. However, all such attempts so far have not had as general applicability as vision. In addition to the practical interest in stereo, the desire to understand the human visual system has led some researchers to propose computational models of the human stereo vision process. Unfortunately, the stereo correspondence problem has proven to be extremely difficult—certainly not as easy as the facility of humans in this problem might suggest. This thesis explores another line of attack, namely the method of differences.

The difficult part of stereo, as for navigation, is the matching problem. In the case of navigation, we desire to compute the camera motion given a set of point distances. The stereo problem is the complement of this in that we wish to compute a set of point distances given the camera motion. In general, the number of points is much larger than the number of parameters of the camera motion. This formulation of the problem shows (roughly speaking) why stereo is harder than navigation: we are given less input and asked to compute more output. That is, the amount of constraint on each quantity to be computed is less.

In addition to navigation and stereo, matching has a number of other applications. For example, one method of approaching object detection in a scene might be to hypothesize the existence of an object in the scene and then to check that hypothesis by attempting to match its known appearance against the picture. In another application, Reddy & Rubin (1978) report the need to align as nearly as possible adjacent slices in lobster nerve tissue, for the purpose of mapping the neuronal connections. Finally, some researchers have looked at the possibility of using object motion detection as a means of compressing the bandwidth required for the transmission of motion picture sequences (See, for example, Limb & Murphy, 1975a, b).

3. Definitions

This section gives a precise definition of the image matching problem, and then generalizes that definition. Then the method of differences is described.

Preliminary definitions.   A few preliminary definitions are in order. The notation used in this thesis is discussed in Appendix A.

An image is a function $I(\mathbf{p})$ of a vector, $\mathbf{p}$. The vector $\mathbf{p}$ denotes a position in the image and $I(\mathbf{p})$ represents the pixel value at that position. For usual images, $I(\mathbf{p})$ is a scalar-valued function; but we consider some "images" in which $I(\mathbf{p})$ is a vector-valued function. I is generally only defined over a bounded rectangular region. Often there will be two images, I_1 and I_2, under discussion.

There will frequently be a need to compare two images, so a metric of image difference will be needed. The symbol E (for error) will denote the difference between two images. The most common such metric, and the one employed in this thesis, is the L_2 norm, defined by

$$E = \sum_{\mathbf{p}} (I_2(\mathbf{p}) - I_1(\mathbf{p}))^2.$$

Here $\mathbf{p}$ ranges over image points in the regions being compared. Other metrics are used in practice, but they are typically generalizations of the L_2 norm.

Matching. The traditional image registration problem can be defined as follows: given two images $I_1(\mathbf{p})$ and $I_2(\mathbf{p})$ related by and $I_1(\mathbf{p}) = I_2(\mathbf{p}+\mathbf{h})$, determine the *disparity* vector $\mathbf{h}$ between them. In many real situations, the stated relationship will not hold exactly, so a slightly different formulation of the matching problem is needed: find a disparity vector $\mathbf{h}$ such that, as nearly as possible, $I_1(\mathbf{p})$ and $I_2(\mathbf{p} + \mathbf{h})$ match. The degree of match is measured by some norm, such as the L_2 norm mentioned above. More formally, we want to find an $\mathbf{h}$ to minimize some measure of the difference between $I_1(\mathbf{p})$ and $I_2(\mathbf{p} + \mathbf{h})$. This form of registration can be viewed as determining two global parameters, namely the components $\mathbf{h}_x$ and $\mathbf{h}_y$ of the disparity vector.

This definition of the matching problem can be generalized in two ways: first, one can model the transformation between the images with more parameters, and solve for those parameters. For example, we could model the change between the images as a linear deformation of coordinates, that is

$$I_1(\mathbf{p}) = I_2(\mathbf{p}A + \mathbf{h}),$$

where A is the matrix describing the linear deformation. (As discussed in Appendix A, we use row vectors, thus we write $\mathbf{p}A$). In this case the parameters being solved for are the translation components $\mathbf{h}_x$ and $\mathbf{h}_y$, plus the parameters characterizing the matrix A: these could be simply the entries of A if it is an unconstrained matrix, or perhaps the rotation angle if A is restricted to be a rotation matrix. In this case one is still solving for a few *global* transformation parameters. As a second generalization, we can solve for a field of *local* image transformations. In this case the image transformation is

$$I_1(\mathbf{p}) = I_2(\mathbf{p} + \mathbf{h}(\mathbf{p})).$$

This equation allows for a disparity $\mathbf{h}(\mathbf{p})$ that is different at every point $\mathbf{p}$ of the image. With the right choice of parameters, the former generalization allows us to do optical navigation, which is the subject of Chapters 3 and 4 of the thesis. The latter generalization allows us to compute a "map" of local disparities from a stereo pair, which gives us information about the three-dimensional form of the scene; this is discussed in Chapters 5 and 6.

The method of differences. We are now in a position to understand the method of differences. The method is based on the assumption that the difference between image intensities $I_1(p)$ and $I_2(p)$ at a point p can be explained, to a linear approximation, by the disparity h between the images and by the image spatial intensity gradient. The relationship is given by

$$I_1(p) - I_2(p) \approx h_x D_x I_2(p) + h_y D_y I_2(p). \tag{-1}$$

Here, h_x and h_y are the components of the disparity vector h, and D_x and D_y denote partial differentiation with respect to x and y. This approximation is discussed further Chapter 2 of the thesis. The method takes its name from the fact that it uses the image intensity differences together with the intensity gradient (which is approximated by differences) to obtain a linear constraint on the parameters being solving for, h_x and h_y.

As equation (-1) shows, each point p results in one linear constraint. Since in this case we are solving for the two quantities h_x and h_y, we will need to combine evidence from at least two points p to obtain a unique solution. In the generalized case we will obtain similar linear constraints on the parameters being solved for, for example the camera motion parameters; we will need as many points as there are parameters. In practice, using a least-squares technique allows combining evidence from many more points than there are parameters, thus reducing the effects of noise and somewhat ameliorating the approximate nature of equation (-1). This is done by minimizing the total squared deviation from the linear constraint of equation (-1), given by

$$\sum_p \left(I_1(p) - I_2(p) - h_x D_x I_2(p) - h_y D_y I_2(p) \right)^2.$$

Minimization of this quantity is straightforward. The set of points p that the sum above ranges over is chosen in one of two ways: if we are solving for global parameters (such as the motion parameters), we combine the information from a set of feature points selected from the whole image. These feature points should be selected to be near edges or other similar features whose position is little affected by photometric error, such as that caused by noise and specular effects. If we are solving for a field of local parameters, we combine the evidence from each point in the neighborhood of a given point to obtain the parameters at that point.

The use of the approximation of (-1) has two implications. First, it means that the computed parameters will only be approximate. This problem is solved by iteration: taking the computed parameters as initial guesses, use the method of differences to compare the images as deformed by those parameters to obtain better guesses. This results in a gradient-based iteration, which under the right circumstances will converge to the correct values of the parameters. One concern of the thesis is the conditions under which convergence is achieved. The second problem is that the linear approximation is a good one only over a certain range. The effect is to limit the range over which the iteration just described will converge. This problem we solve by smoothing the images; as is shown in the thesis, both

in theory and by experiment, smoothing increases the range of convergence, but at the expense of accuracy. Thus we adopt a coarse-fine approach, in which very smooth images are used in the first iteration, and less smooth images in later iterations. The techniques of smoothing and iteration are fundamental to the method of differences.

4. Summary

Chapter 1 lays the groundwork by defining the traditional image registration problem and characterizing it as a parameter estimation problem, where the parameters are the x and y offsets of the matching parts of the two images. Such a characterization opens the door to generalization, as shown above. For example, we could characterize the transformation between the images as an affine transformation of coordinates rather than a simple translation. But more importantly, the optical navigation and the stereo interpretation problems can be cast as parameter estimation problems. Thus, they can be solved directly by the method of differences without first solving an intermediate matching problem.

The theory of image matching by the method of differences in a general context is developed in Chapter 2. This chapter introduces the idea of *global* match parameters (such as the navigation parameters relating two cameras) and *local* match parameters (such as the distance of a point from the camera). In either case, changing one of the parameters changes the position of the matching point. Thus the chain rule allows us to derive linear constraints on general match parameters, much like the linear constraint on translation match parameters in equation (-1). This means that the method of differences can handle generalized matching problems; a variety of such problems are discussed in Chapter 2. The importance of smoothing and iteration to the functioning of the method are discussed. In the second part of the chapter, a theoretical analysis gives support to the intuitive ideas that smoothing and iteration improve the performance of the algorithm. In particular, the theory shows that behavior of the algorithm depends only on power spectrum of the image, and is independent of phase spectrum. This has two implications. First, the power spectra of images tend to be similar from image to image: they fall off with something like and exponential decrease as frequency increases, characterized by one or two parameters. This means that the results obtained with the experimental images will be indicative of results obtained with other similar images. Second, it allows general predictions about how smoothing will affect the performance of the algorithm; the prediction is that it will increase the range of convergence. In addition the theoretical analysis provides a way of predicting of the range and speed of convergence.

Chapter 3 focuses on issues related to one particular type of matching, namely matching for optical navigation. A discussion of the applications of optical navigation comes first. Then a camera model is defined and the parameters relating two cameras are enumerated. These parameters are both geometric (relating the geometry of the cameras) and photometric (relating the intensity values reported by the cameras). The navigation problem is to solve for (some subset of) these parameters. This is done by using the method of differences; the mathematics of this is developed in some detail, and the implementation

of the operations is described. The net result is a system of linear equations, one for each of the unknowns. Of considerable interest is the numerical stability of the linear system of equations in the camera parameters derived in the course of applying the method of differences to this problem. The first step in the analysis of the stability is to separate the geometric from the photometric effects: our intuition is that there are certain geometric conditions on the points under which it should be impossible to determine the camera parameters. A measure of geometric stability is developed that satisfies our intuition in two ways: first, it is closely related to the stability of the actual matrix to be inverted, and indeed experiment shows it is a reasonable predictor of that stability. Second, it can be interpreted as a measure of the correlation of the geometric behavior of the parameters being solved for. If two geometric parameters are highly correlated, it should be difficult to solve for them. Analysis of this measure leads to the conclusion that the equations will be numerically unstable when the reference points have a "flat" distribution in three-space. This can occur because the points are in fact on a flat surface, or are well-distributed in three-space but the scene is viewed from a distance with a long-focal length lens.

With this theoretical foundation, Chapter 4 sets out to verify the theory by experiment. These experiments were conducted on both real and synthetic data. The main results of these experiments are as follows. First, it is verified that the geometric correlation measure is a good predictor of the the stability of the actual equations. Its advantage of course is that it is independent of the actual scene. The experiments are in agreement with our prediction that "flat" point distributions or scenes viewed from far away resulted in ill-conditioned equations. The adequacy of the condition measure depends on the expected effect of noise on the equations; experiments in which random noise is added to the pictures shows that the observed conditioning is satisfactory. The range of convergence for one-parameter estimation is determined to be as follows: for pan and tilt, anywhere from ± 10 degrees to ± 50 degrees, depending on the scene; for roll, up to ± 30 degrees independent of the scene. For the position parameters $(x, y,$ and $z)$ the tolerance is as much as ± 1 meter in a room-sized scene. These ranges of course depend on the degree of smoothing; they amounts mentioned here correspond to very large smoothing windows, as much as $\frac{1}{4}$ the image size. A wider angle view allows larger smoothing windows and so more tolerance, but decreases accuracy. In the multi-parameter case the range of convergence becomes smaller as the number of parameters is increased, but retained a useful range even in the six-parameter case. The accuracy observed is essentially that expected for 1-pixel accuracy in the individual matches. For typical images, accuracy is about ± 1 or 2 cm in x and y; less accuracy is obtained in z. Angular accuracy is extremely high. Finally, the method involves accumulating a few dozen quantities per reference point per iteration, and so is quite fast. These properties also make it suitable for implementation on special-purpose hardware. These times do not include the time required to smooth the images, but hardware techniques for doing so are well-understood and available in commercial units.

Chapter 5 discusses the version of the algorithm to be used for computing a stereo disparity map. While a navigation algorithm seeks to estimate a few global parameters, a stereo algorithm computes a field of local parameters; each parameter is the local displace-

ment of the image due to binocular disparity, and is a direct measure of the distance to the point. The approach taken is to use the method of differences to match a small image patch around each given point against the other image. This process is repeated iteratively, yielding better and better disparity estimates. Actually, the match is at each iteration is done against the other image as distorted by the current disparity field estimate. As the process is iterated, successively less smooth images and smaller matching windows are used. This allows correct matching to take place even though the images may distorted relative to each other due to perspective. An algorithm based on an efficient technique for image smoothing (described in Appendix B) allows this to be done efficiently and in time independent of the window size. The method requires about 30 sec per iteration for a 250×250 image on a VAX 11/780 (independent of smoothing time). Its regular structure makes it quite suitable for implementation on special-purpose hardware.

Chapter 6 presents the results of experiments designed to test the stereo algorithm. Again, both synthetic and real data are used. The synthetic data consist of random-dot stereograms, both of the "floating-square" type and of the "rolling-hills" type. Excellent disparity maps are obtained in both cases. The real data consist of aerial views of Washington, D.C. Promising results are obtained: buildings and streets are detected. An experiment to determine whether matches provided by an independent source (another stereo matching program) could improve the performance of the method-of-differences algorithm are inconclusive. The problem with evaluating the results on the real images is the lack of ground-truth data against which to judge them, or of a vision system to take disparity maps as input for some image understanding task. Nevertheless, the results seem promising.

Finally, Chapter 7 summarizes the thesis, discusses its main contributions, and suggests avenues for further research. In addition, Appendix A discusses the notation used in the thesis, and Appendix B presents the algorithms used for smoothing and gradient estimation.

5. Conclusions

This thesis investigates the usefulness of the method of differences as an image matching technique. In particular the method is applied to two problems: optical navigation and stereo vision. We see that the technique is applicable in any situation where a (very) rough estimate of the match is available, but an accurate answer is desired. A substantial part of the research is directed toward determining how rough the estimate may be and how accurate the final answer is. It is demonstrated that that the method has adequate range and accuracy for many robotic tasks, particularly as applied to optical navigation.

This research reveals several factors that are essential to making the method of differences work. We see the importance of smoothing and of iteration to the method. Roughly speaking, without smoothing the technique has too little range and without iteration it has too little accuracy to be useful in most applications. Because smoothing reduces the accuracy of the method, different degrees of smoothing can be used at each iteration, to yield a coarse-fine method. Moreover, the thesis recognizes the the importance of using points near intensity edges because of the relative unambiguity of matches for such points. This

fact has long been recognized by advocates of edge-based processing techniques. However, as this research shows, the importance of points near edges does not demand edge-based algorithms.

The method provides several advantages over other matching techniques. It is free of search, which can be impractical because of its expense in multi-dimensional parameter spaces. With a standard search, the expense goes up like the size of the volume to be searched, which is as the power of the dimensionality of the parameter space; whereas the expense of the method of differences is roughly a function only of the distance of the initial estimate from the actual answer. Thus, the method of differences has its greatest advantage in high-dimensional parameter problems, such as navigation. However, it provides an advantage even in low-dimensional spaces, such as the disparity in a stereo depth map. This is because the structure of the algorithm allows for an efficient implementation, as we saw in the computation of the depth map. Furthermore, for parameter estimation problems it computes exactly what is needed without proceeding through intermediate results, such as point matches or an optical flow field. Finally, its regular and simple structure make it quite suitable for future implementation on special-purpose hardware.