



University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

December 1981

A System for a Total Matching of Sterio Pairs of Images

Lucio de Risi
University of Pennsylvania

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Lucio de Risi, "A System for a Total Matching of Sterio Pairs of Images", . December 1981.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-81-15.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/644
For more information, please contact repository@pobox.upenn.edu.

A System for a Total Matching of Stereo Pairs of Images

Abstract

A system which provides a total matching of a stereo pair of images is described: to every point in the first image a corresponding point in the second image is assigned. The processing consists of determining a limited number of matching candidates for each point in the first image and ordering the possible matches of a point for decreasing value of likelihood. Geometrical constraints are applied to determine the consistency among matches associated with different points. The assumption that points close in the image correspond to points close in space (continuity assumption) has been used. In most of the cases the scenes are composed of objects in contrast with the background. The target image is then preprocessed in order to extract the objects and then to reduce the areas to be matched to the object areas.

The algorithm has been successfully applied to artificial scenes and real scenes. Examples and results are presented.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-81-15.

UNIVERSITY OF PENNSYLVANIA
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

A SYSTEM FOR A TOTAL MATCHING OF
STEREO PAIRS OF IMAGES

Lucio de Risi

Philadelphia, Pennsylvania
December 1981

A thesis presented to the Faculty of Engineering and Applied Science of the University of Pennsylvania in partial fulfillment of the requirements for the degree of Master of Science in Engineering for graduate work in Computer and Information Science.


Prof. Ruzena Bajcsy

Prof. Aravind K. Joshi

ABSTRACT

A system which provides a total matching of a stereo pair of images is described: to every point in the first image a corresponding point in the second image is assigned. The processing consists of determining a limited number of matching candidates for each point in the first image and ordering the possible matches of a point for decreasing value of likelihood. Geometrical constraints are applied to determine the consistency among matches associated with different points. The assumption that points close in the image correspond to points close in space (continuity assumption) has been used. In most of the cases the scenes are composed of objects in contrast with the background. The target image is then preprocessed in order to extract the objects and then to reduce the areas to be matched to the object areas.

The algorithm has been successfully applied to artificial scenes and real scenes. Examples and results are presented.

ACKNOWLEDGEMENTS

I am grateful to Dr. Ruzena Bajcsy for the advice and assistance provided in this thesis. In addition, I am indebted to her for making it possible for me to carry out this research in a relatively unconstrained academic environment.

I would also like to thank Mr. Inderjeet Mani for his help and criticism in reviewing this thesis and Mr. Gopalan Nadathur for his bountiful assistance in solving mathematical problems.

Finally I wish to thank the Fulbright Commission for the support and assistance provided during the year spent at University of Pennsylvania.

This work has been partially supported by National Science Foundation Grant MCS 81-96176.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	page 1
CHAPTER 2: STEREO SCENE ANALYSIS	5
2.1. Image Acquisition	6
2.2. Camera Calibration	7
2.3. Feature Acquisition	10
2.3.1. Point-like Features	11
Moravec Operator	11
2.3.2. Edge Detection	12
Rosenfeld Algorithms	14
Zero Crossings	15
The Use of Hough Transformation	16
2.3.3. External Contours	17
The Use of Relaxation	18
2.4. Matching	19
Grimson Algorithm	23
Barnard and Thompson Algorithm	26
CHAPTER 3: OBJECTS-BACKGROUND SEPARATION	29
3.1. Description of the Algorithm	29
CHAPTER 4: THE MATCHING ALGORITHM	37

4.1. Introduction to the Matching Algorithm	37
4.2. Implementation of the Matching Algorithm	40
4.3. Example	44
4.4. More Details about the Algorithm and its Application to Stereo Pairs of Images	48
APPENDIX A: CHECKING FACTOR	54
BIBLIOGRAPHY	60

CHAPTER 1

INTRODUCTION

In the Image Understanding environment a stereo scene analyser is a system dedicated to obtaining depth measurements about the elements of a scene, from two or perhaps more images taken from different viewpoints. The geometrical problem which underlies almost the entire stereo analysis is the possibility of reconstructing the three dimensional representation of a scene from two two dimensional images.

Let us suppose in the simplest case that we want to determine the position in space of a single point P (fig. 1.1). If we know its projection p_1 on an image plane I_1 and if we know the location of the center of projection C_1 , we will be able to define the straight line in this space on which the point must lie. This straight line, called the projecting ray, is the one connecting C_1 with the projection p_1 of the point on the image plane. Simple monocular detection yields no more information and gives to the problem a solution of the type many-to-one. Although a point in the space has a unique projection on any image plane, an image point is associated with a line in the space and it can be the projection of any point which lies on it.

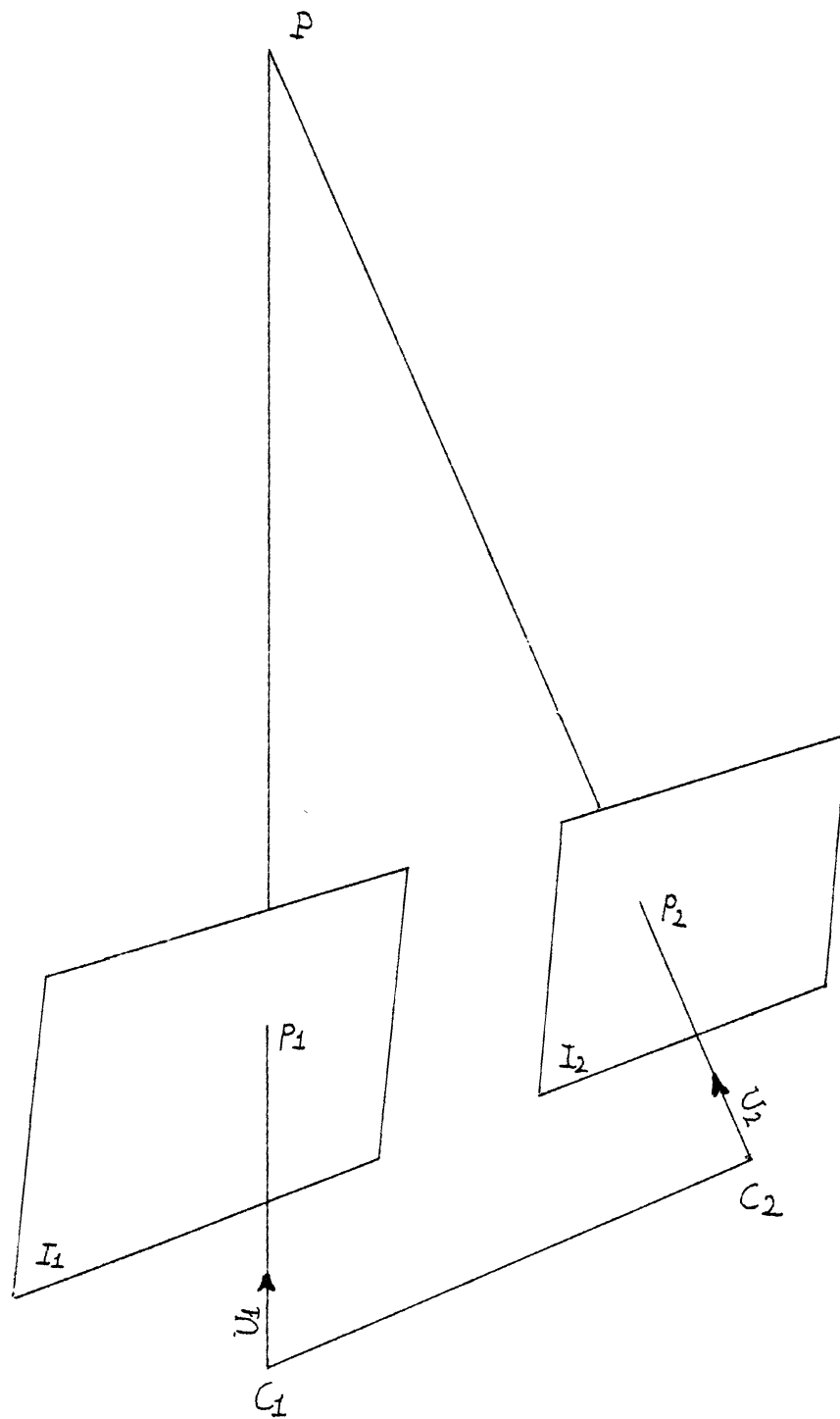


fig. 1.1

If we now suppose that we are given another projection p_2 on I_2 of the same point taken from a different position C_2 , we can correctly locate the point P on a second straight line and therefore determine, from the intersection of the two projecting rays, the position of the point in the space. The computation needed to find the intersection is straightforward. If U_1 and U_2 are the unit vectors of the rays projecting P on I_1 and I_2 respectively and C_1C_2 is the vector connecting the two lenses, the problem consists of finding two numbers "a" and "b" such that

$$aU_1 = C_1C_2 + bU_2.$$

Here we are merely applying the rule for the addition of two vectors which says that the sum of two vectors is given by the vector whose endpoints are the start point of the first one and the final point of the second one. Either aU_1 or bU_2 will finally give the correct position of the point in the space. In practice, because of the non-continuous resolution of the image planes and because of various other errors, the above equation will never have a solution and a reasonable approach to the problem is to find two numbers "a" and "b" such that they minimize the square of the difference [5]:

$$[aU_1 - (C_1C_2 + bU_2)]^2$$

In the real applications of a stereo system a scene will very rarely be formed by a single point and much more complex situations are analysed. The above discussion is, anyway, applicable to each pair of corresponding image points, no matter how complex the scene we are dealing with may be.

CHAPTER 2

STEREO SCENE ANALYSIS

So far we have stated the geometrical basis for a stereo analysis, but no attention has been paid to how two corresponding image points are identified, what type of points is more useful to consider in order to reconstruct the scene, and how the images are obtained. Answering these questions is equivalent to implementing the various parts of a stereo scene analyser. In the following we shall discuss these parts, that is, we shall describe the problems related to:

- a. Image Acquisition
- b. Camera Calibration
- c. Feature Acquisition
- d. Matching

We are particularly stressing on the feature acquisition problem and on the matching problem which are the crucial points in computational stereo and of major interest for the development of this thesis.

For a survey of computational stereo the reader can refer to Barnard and Fischler [2].

2.1. IMAGE ACQUISITION

By image acquisition we refer to the problem of obtaining a digitized image of the projection of a scene along a certain direction. In digitizing an image two very important factors, which will strongly influence the subsequent processes, can be identified:

- a. Resolution
- b. Level of noise

A high resolution image has the advantage of enhancing the details of the scene, therefore objects can be represented without losing a lot of information about them and algorithms can be applied more successfully and they can also be more sophisticated. The disadvantage of high resolution images is that more computation is required and as a result the cost of processing the image increases. Both high resolution and low resolution images have their fields of application as discussed in [2].

The level of noise is a crucial point. The results of both feature acquisition and matching algorithms are most often affected by the presence of noise. Since this generally appears in the form of anomalous points which introduce high frequency components into the spectrum of the image (Fourier transform), a low-pass filter is a method for re-

moving at least part of the noise from the image. Naturally, if a low-pass filter is not applied cautiously, a blurred image will result with consequent loss of information especially in those areas which contain edges or other intensity features and which are therefore meaningful for understanding the picture.

In the space domain a low-pass filter is implemented by averaging algorithms which are applied around each image point. Windows of different size and shape can be used and weights can also be assigned to the various points in the window obtaining different results.

2.2. CAMERA CALIBRATION

In this section we shall discuss, for the sake of completeness, the role of camera calibration in designing a stereo scene analyser. We shall describe the general problem without going into the details of the implementation since this has not been subject of this thesis. We shall conclude the section giving some references for the latter.

As we have already discussed earlier, a stereo system uses the two dimensional information coming from the projection of a scene in two different image planes to re-

construct the three dimensional representation of it.

When we talked about the projection of a point in an image plane we were implicitly assuming that its position in the plane was determined with reference to a two dimensional coordinate system with the origin somewhere in the plane. The most natural reference is a orthogonal cartesian system centered in the intersection of the main projecting ray with the plane, where the main projecting ray is the straight line from the center of projection perpendicular to the plane. Now, a stereo system has to carry out some computations using the coordinates of corresponding image points, that is it has to apply the inverse perspective transformations, to determine the location of the point in the space. In order to make this possible the coordinates of the image points and of the point in the space must refer to a common coordinate system. This can be one of the image plane reference frames extended to 3-D by considering an axis coincident with the main projecting ray or it can be any other reference independent from the image planes. We shall call the latter a world coordinate system. To consider this for the three dimensional representation of points seems to be more appropriate since the independence from the image planes lets the reference be located wherever we think is preferable.

In any case a change of coordinates is required and some parameters to make it possible have to be calculated. In other words, the translation of the origin and the three rotations around the axis needed for moving from a frame to another are to be determined. If we assume that the image planes never rotate around the main projecting ray, the problem is to determine the components X_o Y_o Z_o of the translation, the pan angle and the tilt angle.

If we went through the tedious calculations of direct perspective transformations with two reference frames, we would find out that an image point given in image plane coordinates is related to the corresponding object point given in world coordinates by a function of the parameters which define the change of coordinates and of some other parameters related to the camera. Even if these parameters could in principle be directly computed, it is generally more convenient to derive them using the camera itself. This leads to the camera calibration problem.

The idea is that if a set of object points and a set of corresponding image points are given, the parameters can be determined to be the ones which minimize the distance between the image points computed from the object points using the perspective transformations and the given image po-

ints. Although this seems to be just a matter of solving a system of equations, the solution is not straightforward. First the function which relates image coordinates and world coordinates is sufficiently difficult not to suggest a direct solution. Secondly we cannot expect that by matching a set of computed points with a set of given ones in a discrete picture we will easily achieve an accurate solution for the parameters.

For a better understanding of the problem the reader can refer to Duda and Hart [5], while for examples of practical implementations we cite Gennery [8], Yakimovsky and Cunningham [19], Fischler and Bolles [7].

2.3. FEATURE ACQUISITION

Feature acquisition is the problem of finding in an image significant elements which can be helpful for understanding the scene and for further processing it. By significant elements we mean any part of the image, an area or even a point, which holds some interesting properties that allow us to characterize it in the context of the scene in the best possible manner. These properties are related to particular grey values or to high variations of them. In the following discussion we will give a possible classifica-

tion of the features which are used for image understanding and then illustrate some algorithms which have been implemented to derive them.

On the basis of their different characteristics, features can be classified as:

- a. Point-like features
- b. Edges
- c. External contours

2.3.1 POINT-LIKE FEATURES

Zones of high variance are, intuitively, areas in which a point-like feature can be localized. Given a window of a fixed size, if the variance inside it exceeds an assigned threshold, we can assume that the center of the window is an interesting point. However, a point with high variance around it is really an interesting feature if the variance is not unidirectional. In fact, a high variance in only one direction does not permit the separation of the point from the others which lie on a line perpendicular to that direction.

MORAVEC OPERATOR

Moravec [15] has designed an algorithm which takes

care of this problem by computing the sum of the squares of the differences between adjacent pixels, in a small window of fixed size (4*4 in the implementation). The sum is computed in four directions: horizontal, vertical, the two diagonals. A window contains an interesting feature if the lowest of the sums calculated over the window is a local maxima among the 25 windows overlapping or contacting it. The 25 windows are obtained by shifting along the horizontal by 2 or 4 pixels and/or along the vertical by 4 pixels. In order to reduce the influence of the noise and the computation time, the features are determined in an image reduced by a factor of 2. This algorithm plays an important role in obtaining the information for driving a robot rover who reasons visually.

2.3.2 EDGE DETECTION

The easiest way to have an indication of the presence of an edge through a point (x,y) in a picture is to compute in that point the one dimensional derivative of the function $g(x,y)$ describing the image, i.e.:

$$g(x,y) - g(x',y')$$

where (x',y') is in the 8-connected neighborhood of (x,y) .

For example, if $x'=x+1$ and $y'=y$ we can rather easily detect the presence of a vertical edge: a high absolute value of the derivative will be connected to the presence of an edge.

The natural extension of the above operator to the two dimensional case is the gradient:

$$\text{abs}[g(x,y)-g(x+1,y+1)] + \text{abs}[g(x,y+1)-g(x+1,y)]$$

known as Roberts' operator. Improvements for both the one dimensional derivative and the gradient can result from computing averages over one or two dimensional windows around the point respectively and therefore using the averages to feed the operators. This expansion has the advantage of greatly reducing the influence of the noise.

However, even if the influence of the noise can be reduced by averaging over neighborhoods of the point, for a higher level of noise and blurring and for textured images, good performances of an edge detector require more sophisticated algorithms. These algorithms use windows of different size. Small windows are more sensitive to variations, locate the edges better but are also very sensitive to the noise. On the other hand, large windows average the noise but are less sensitive to variations and produce thicker edges because they respond to variations in wider areas.

ROSENFELD ALGORITHMS

Rosenfeld has proposed some algorithms based on windows of different size [16,17]. He computes the differences between average grey values of pairs of windows around the point. Windows of different size are used in different iterations and then the results are multiplied to obtain the edge information. In doing so, if all the differences are large, i.e. if major edges are present, the final product will be large and the edges will appear very sharp. However, since the product includes all the differences, minor edges will also be detected. In another algorithm he defines the size of the window to apply as a function of the point under consideration; for each point the largest possible window is used. The size is determined when, at a certain level of expansion of the window, the differences decrease sensibly with respect to the previous size. In practice, for major edges or for minor edges near major edges the expansion will stop at high values since the differences will be constantly high; for isolated minor edges the expansion will stop soon saving the information about minor edges; for micro-edges due to the texture the differences will be constantly low. This way the noise near major edges will not be detected. Eventually, in order to detect edges in between areas with similar texture but of

different coarseness, the image can be preprocessed with a small window obtaining in output grey levels depending on the coarseness of the texture, and then the above algorithm can be applied again.

ZERO-CROSSINGS

In accordance with the Marr and Poggio theory [13] of human stereo vision, Grimson [9] uses zero-crossings as features to be matched. A Gaussian low-pass filter is first applied to the image using circular neighborhoods; this means that the image is averaged and that the average value of each point is given by the points around it weighted over a Gaussian surface. To the resulting image the Laplacian, which is a non directional linear second derivative operator, is applied. Four circular neighborhoods of different size are used for the above process. Sharp changes in the intensity picture will correspond to peaks of the first derivative and therefore to zero-crossings of the Laplacian. The advantage of using windows of different size is that variations of different levels can be detected. Windows of smaller size allow the detection of smaller changes but are also more influenced by the noise; larger windows average the noise much better but can also average small variations which would not be picked up by the operator. After the operator has been applied over the whole image, zero-crossings

are determined by horizontally scanning the image. When a change in the sign is encountered a zero-crossing is identified.

THE USE OF HOUGH TRANSFORMATION

An interesting way for detecting lines and curves of a known equation is to apply the Hough transformation [6]. In particular, if we want to detect straight lines, a very simple algorithm can be implemented. Since a straight line is unequivocally determined by two parameters, a correspondence between a line and a point in a parametric plane can be established. For instance the distance R , computed perpendicularly to the line, between the line and the origin of a Cartesian plane and the angle A that R forms with the x -axis can be used as parameters. A is in the range $[0,180]$. Now, if a point (x',y') is represented in the parametric plane R,A by the sinusoid of equation:

$$R=x'\cos A+y'\sin A$$

it is clear that any point of the sinusoid represents a straight line through the point (x',y') . In fact, given a point (R',A') of the sinusoid this has associated with it, in the Cartesian plane, the straight line of equation:

$$R'=x\cos A'+y\sin A',$$

which is certainly satisfied by (x',y') . Now, if for each point possibly on an edge we consider all the straight lines (R,A) through it and we record them, we will end up with knowing how many possible edge points are on each line: the number of points will equal the number of times that line has been recorded. Finally only the lines which have a number of points exceeding a fixed threshold will be considered as edges.

Many other algorithms for edge detection have been proposed; we here cite Herskowitz and Binford [11], Heuckel [12], and, for sequential edge detection Martelli [14] who takes the edge detection problem into a graph search.

2.3.3 EXTERNAL CONTOURS

An algorithm for finding the external contours is a particular edge detector algorithm which restricts the edges to being the external contours of an object. We discuss this aspect of edge detection separately from the general case because of two factors that we think characterize external contours in the image understanding context. First an algorithm for finding external contours can be applied only on a scene constituted of few well defined objects in contrast with the background (block's world-like

environment); secondly the only purpose of such an algorithm is to separate the interesting part of the scene (objects) from the background, in order to allow future processes to polarize their attention only on part of the image.

THE USE OF RELAXATION

Danker and Rosenfeld [4] have presented an algorithm which uses relaxation for blob detection. A blob is a region which is in contrast with the background, i.e. a compact region lighter or darker than the background. They use two different pieces of information to determine the blob area: the probability that there is an edge through a point and the probability for a point of being on the interior of the blob or on the exterior of the blob. The initial estimate of edge probability is given by $p=e/E$ where E is the largest value resulting from the application, at each point, of Prewitt operator in 8 directions at 45i degrees; and "e" is the largest value at the point under consideration, say P . Moreover, the estimate of the probability that there is an edge in direction "i" is given by $(e_i/s)*p$ where "e_i" is the value of the operator for direction "i" at point P and "s" is the sum of all "e_i" in the 8 directions. The initial interior/exterior probabilities are computed considering the lightest and the darkest point in the picture. If "g" is the grey value at P , "l" and "d" the largest and smallest

grey values in the picture, the probability for P being light is given by $p=(g-d)/(1-d)$ and the probability for P being dark is $1-p$. These probabilities are then updated in different iterations using criteria of compatibility between labels of neighbouring points. A label is in the first case "to have an edge in a certain direction" or "to have no edge", in the second case "to be an interior point" or "to be an exterior point". The authors have implemented the algorithm using only the information about edges, only the interior/exterior information and finally both types of information together. They have shown that, when the entire information is used, the number of iterations needed by the algorithm to perform the bulb-background separation decreases.

An algorithm for separating objects from the background has been implemented for the purpose of this work and it will be illustrated later in detail.

2.4. MATCHING

Given two image planes I1 and I2 and the projection of a point P in one of them, say I1, the problem of finding the projection of P in the other plane is called the "correspondence problem" or matching problem.

Let p_1 be the projection of P in I_1 , p_2 be the projection of P in I_2 , C_1 and C_2 be the centers of projection (fig. 1.1). Because of the geometry of a stereo system, the search for the match (p_2) is constrained to the line intersection of the image plane I_2 with the plane defined by C_1 , C_2 and the projection p_1 of P . This plane is called the epipolar plane and the intersection with I_1 or I_2 is an epipolar line. Image I_1 is also called the target image and image I_2 the candidate image.

The most widely used method to give an estimate of the location of the match in I_2 is to compute the discrete normalized correlation between an area in image 1 centered in p_1 and an area in image 2 centered in each point which is a match candidate [10]:

$$\text{CORR} = \text{SUM}(G_{i1} * G_{i2}) / \text{SQRT}[\text{SUM}(G_{i1} ** 2) * \text{SUM}(G_{i2} ** 2)]$$

In the above formula G_{i1} and G_{i2} are the grey values, say g_{i1} and g_{i2} , around p_1 and p_2 computed normalized to the means M_1 and M_2 :

$$G_{i1} = g_{i1} - M_1 ; \quad G_{i2} = g_{i2} - M_2 .$$

When the correlation reaches the maximum value, the corresponding point in I_2 will be assumed as the most probable match. This form of the correlation has been used since it

takes care of two problems that may arise when searching a match in I2. First an area in I2 can equal the corresponding area in I1 except for an offset in the grey value, that is $gi2=gi1+OFFSET$; secondly image I2 can have a gain Q with respect to image I1 which would determine, with the above offset, a relation of the type

$$gi2=Q*gi1+OFFSET.$$

Subtracting the means eliminates the problem of the offset, dividing by the variances balances a possible gain.

A problem in computing correlation is the size of the window. A small window speeds up the computation and gives high values of the correlation since in small areas there are less inequalities; a large window is computationally expensive and gives lower values of the correlation but it has the advantage that a large window around a point which is not a match will have less likelihood of resembling the window around the target point, so reducing the probability that a wrong match will correspond to the highest value of the correlation.

In any case the correlation measure or another similar measure of the location of a match are not alone enough to solve the correspondence problem with confidence and some more sophisticated strategies have to be devised. A first

solution to the problem is to use more local information, that is one can compute different measures in the neighbourhood of the target point and of its potential matches and then compose them together to determine a degree of similarity between the target and its potential matches. A high similarity will support the choice of a match.

An alternative solution or an improvement to the latter can be obtained by considering the possible matches of the point in question and of its neighbours and using some criteria for checking the consistency among them and biasing the choice of a match, on the basis of the matches assigned to the neighbours. Checking means that, given two neighbouring points and two possible matches of them, we want to ensure that these matches can co-exist with respect to some geometrical or logical constraints. Biasing the choice of a match means accepting that a point is assigned a certain match depending on how well this choice is supported by the potential matches assigned to points neighbouring the one under consideration.

This last technique is called "Relaxation Labeling". Both checking and biasing are based on the assumption that points close in an image are projections of points which are close in the space too. This assumption is known

as the "continuity assumption". Sharp changes in the depth of adjacent points would make the above strategies fail.

In the last few years many algorithms for solving the correspondence problem have been proposed. In the following, with no attempt at being exhaustive, we will illustrate two of them (Grimson [9], Barnard and Thompson [3]) which can give a good flavor of different approaches to the problem. The reader can refer to Baker [1] for a characterization of edges to be matched. The edges are obtained as zero crossings of the one-dimensional second derivative operator.

GRIMSON ALGORITHM

As we have already said while discussing edge detections, Grimson uses zero-crossings as features to be matched. A zero-crossing is characterized by 3 attributes which are important for determining a match: the channel which has computed the zero-crossing, the sign and the orientation. These attributes can be explained as follows:

- each image is processed using 4 windows of different size (see discussion for feature acquisition) simulating the behaviour of the optical channels in a human stereo system and therefore a zero-crossing has to match with a zero-crossing in the other image computed with the same

window;

- the sign is related to the fact that a zero-crossing can result from passing from a positive value to a negative value or viceversa;
- the orientation is the direction, computed with increments of 30 degrees, of the gradient of the Gaussian filtered image at that zero-crossing.

A necessary condition to accept a zero-crossing in the second image as a match of a zero-crossing in the first image is that the above attributes are equal.

For each zero-crossing an area around the same position in the other image is determined. This area is divided in three parts (pools): a small central one in between two larger pools which expand going away from the center. To each part a disparity is associated: positive, zero and negative. If a zero-crossing has a match in only one of the pools, the match is accepted and recorded; if it has two matches in two different pools, the matches are marked as ambiguous; if it has two matches in the same pool no match is assigned.

At this stage it can be interesting to observe that, because of the way the zero-crossings are characterized, when there are more candidates for the 'same zero-crossing,

there is no information associated with them for computing a measure as to which of the matches can be assumed to be more probable. Two zero crossings cannot be compared along dimensions such as being positive or negative, having the same direction, or being from the same channel. Some indications have to be derived from the matches already assigned to the neighbours in order to eventually choose among the possible candidates. As we will see in the second example we present, an alternative technique can estimate an initial probability that a candidate match is correct and then apply iterative algorithms to modify these probabilities on the basis of the neighbouring evidence, until a situation in which the final decision can be made is reached.

An attempt to disambiguate the possible matches of a zero-crossing is made by considering the matches of the neighbouring zero-crossings and then computing their predominant disparity sign. If one of the possible matches under consideration holds that disparity, that match is accepted. The next step in the algorithm is to check for regions in which the density of zero-crossings matched is below a fixed threshold. When this happens all the matches in that region are discarded. The entire process above is repeated for all the channels.

Now, since a fine resolution requires matches from images processed with the smallest window and since the pools assigned for finding a match are in this case also the smallest, it may happen that larger channels have matches in regions where smaller channels do not. In this case the disparity information from the bigger channels is used as a cue to take, along different directions of projection, another pair of images in which those regions can be matched. This simulates the adjustment of the eyes in the human visual system. At the end of the process the disparities coming from the smallest possible channels for each region are recorded. Since the disparity information can come from different pairs of images, the different positions of the "eyes" (camera calibrations) are required in order to store all the information together.

BARNARD AND THOMPSON ALGORITHM

Barnard and Thompson use interesting points found by the Moravec operator, described earlier, as features to be matched. After finding interesting points over the whole image, those which do not have an associated value as computed by the operator, above a fixed threshold, are discarded. The operator is applied to both the images and then, for each interesting point in image 1, an association is made with all the interesting points in image 2 which are in

a fixed range about the same position.

For the disparity labels resulting from the pairs consisting of an interesting point in image 1, say P, and one of its potential matches in image 2, an initial probability of being the correct one, is computed. In order to derive this probability the square of each difference between a point in a window around the potential match and its corresponding point in a window around P is calculated; and then the squares are added together to yield the sum, say "s". To each disparity label the weight:

$$w(L)=1/(1+c*s(L))$$

is associated, where "s" is the above sum and "c" is a constant fixed to 10 in the implementation. From these weights the probability that the point is not matchable is defined as:

$$u=1-\max[w(L)]$$

and the initial probability of having label "L" is given by:

$$p(L)=p(L/M)*(1-u)$$

where $p(L/M)$ is the conditional probability that point P has label "L" given that the point is matchable. The conditional probability is given by $w(L)$ divided by the sum of all

weights associated with the other possible labels.

In order to update the probability that P has label "L", all the interesting points in image I within a fixed distance from P are considered. For each of these points the probability of the labels which do not differ from the label in question being more than 1 are added together. The new probability p' is given by:

$$p'(L) = p(L) * (A + B * S(L))$$

where S is the sum just described. A is needed in order not to suppress very low probabilities, B defines how fast the algorithm will converge. $p'(L)$ is finally normalized to let it be in the range [0,1]. Normalizing means dividing by the sum of the probabilities $p'(L)$ of all the labels. The algorithm is stopped when it reaches a stable state or in any case after a fixed number of iterations. The labels which hold a probability exceeding 0.7 are assigned as good matches.

CHAPTER 3

OBJECTS-BACKGROUND SEPARATION

In a scene composed of a limited number of objects contrasting with the background, an algorithm which separates the objects from the background is certainly useful for the purpose of understanding the scene. In fact, it restricts the area of interest on which further processing has to be done; this implies that the computation time is reduced. Moreover, if the background is uniform and it is the correspondence problem that is to be solved, a matching algorithm is prevented from trying to assign matches to flat areas in which differences between points are very smooth. In the following we shall describe an algorithm for the external contours of one or more objects in a picture. Since the objects we will be dealing with are assumed to have a high level of variation in order to create interesting patterns to be matched, the basis of the algorithm is that a small background area will be a smooth area, while a small area partly or completely on the object will be subjected to wider variations.

3.1. DESCRIPTION OF THE ALGORITHM

Given the above assumptions, in order to separate the object(s) from the background, we assign to each pixel in the picture an initial probability of it being the projection of a background point; this probability is dependent on the variation existing in a small area centered around the pixel. An estimate of the variation is actually computed by using the standard deviation:

$$s = \text{SQRT} [\text{Sum}(g_i^{**2}) - \text{Sum}(g_i)^{**2}/n]$$

where "g_i" is the grey value of a pixel and n is the number of pixels to which the computation is extended. If "m" is the maximum value of the standard deviation over the picture and "s" is the standard deviation around a generic point, the probability "b" that this is a background point is given by:

$$b = 1 - s/m$$

and then the probability "o" that it is an object point is given by:

$$o = 1 - b = s/m$$

The probabilities are then normalized to the grey range to form an image in which, by convention, darker pixels are most probably projections of object points and lighter pix-

els those of background points.

Note that points with high probability of being in the background are very likely in it. The converse, however, is not true; conditions like noise create non-homogeneity in background areas but there is no condition causing homogeneity in non-homogeneous areas. Fig. 3.1.b, 3.2.b and 3.3.b show the initial probabilities.

In the resulting images we can see that, due basically to the noise which affects the pictures, some points with high "o" probability will appear within the background. In order to achieve a clear separation between the background and the objects, i.e. to derive well-defined and homogeneous light and dark areas in the picture presenting the probabilities, the following algorithm is applied iteratively.

- First we apply a scan line algorithm which is based on the assumption that the minimum dimension of the projection of an object in the image will always be above a certain threshold T_1 , say 5 to 10 pixels. This allows us to say that clusters of dark points below the above threshold cannot correspond to object points. Another threshold (T_2) is given to the algorithm in order to determine the probability above which we assume that a point belongs to

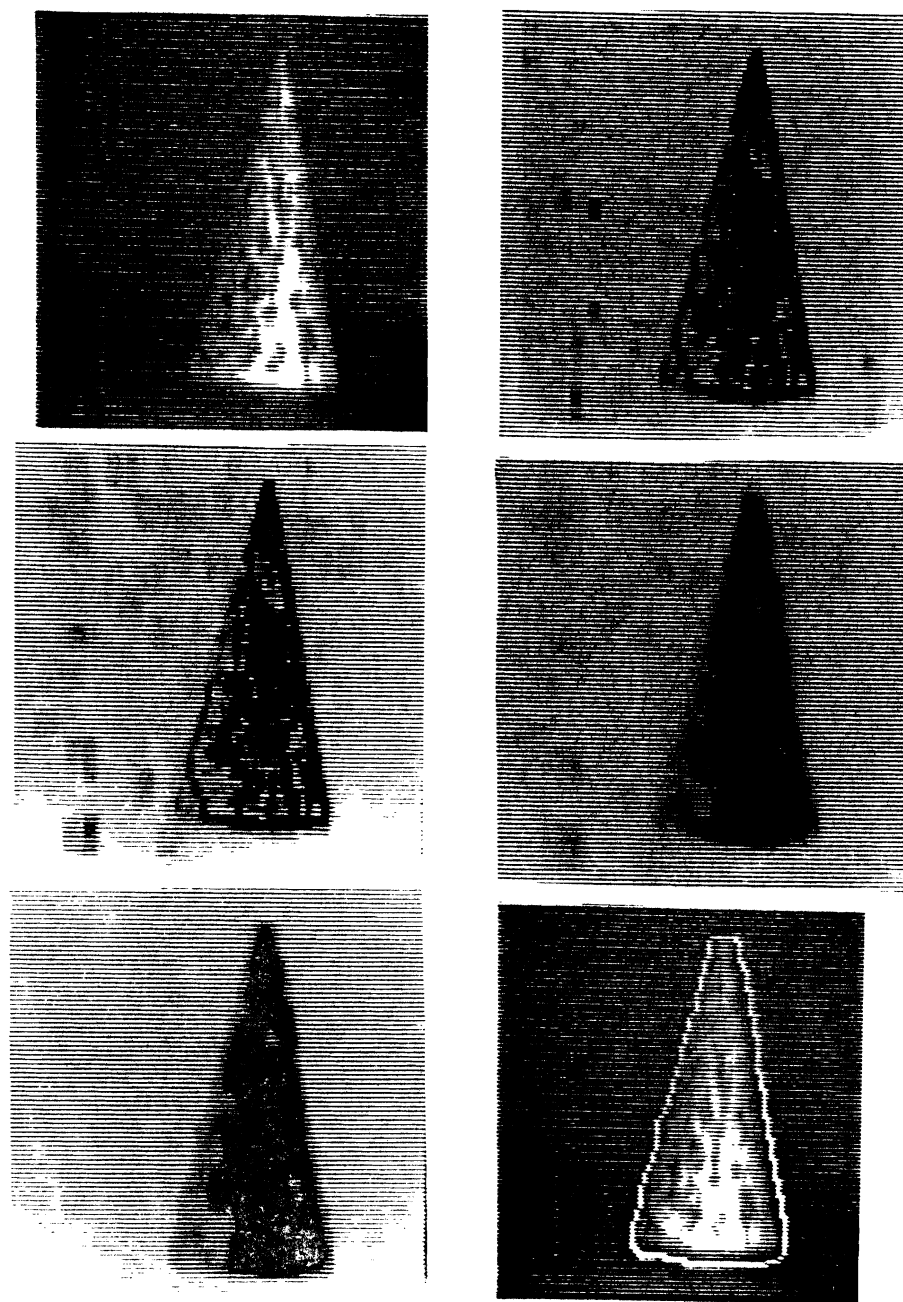


fig. 3.2
OBJECT-BACKGROUND SEPARATION.

- a. picture of the cone;
- b. picture of the probabilities: light areas have high "b" probability, dark areas have high "o" probability;
- c. the probabilities updated after step 1;
- d. update after step 2;
- e. update after step 1 again;
- f. the contours drawn on the original picture.

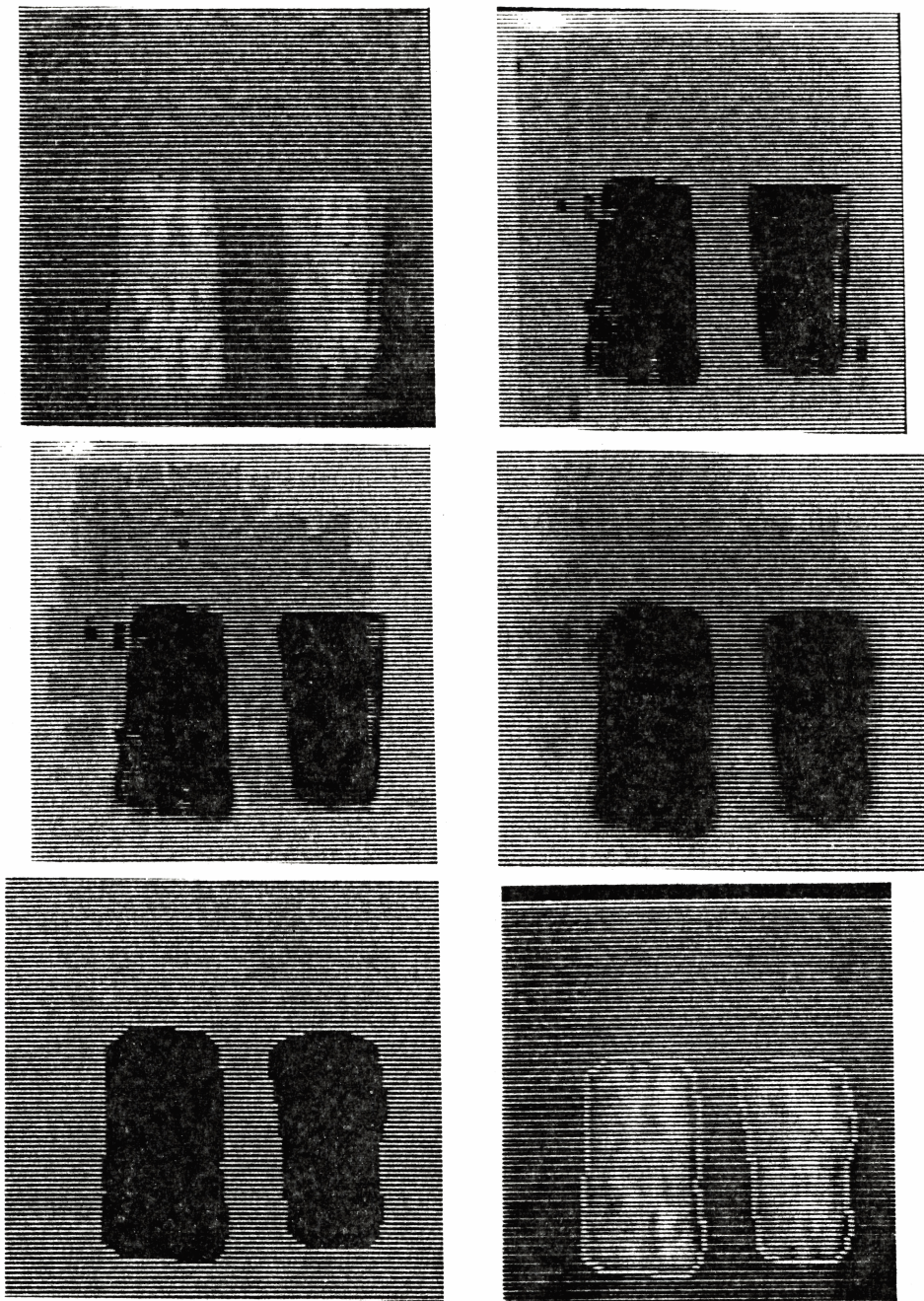


fig. 3.3
OBJECT-BACKGROUND SEPARATION.

- a. picture of the CUPS;
- b. picture of the probabilities: light areas have high "b" probability, dark areas have high "o" probability;
- c. the probabilities updated after step 1;
- d. update after step 2;
- e. update after step 1 again;
- f. the contours drawn on the original picture.

the background. This threshold is first chosen high enough to avoid labelling object points as background. Good values are from .90 to .95. The picture is then scanned line by line and column by column and each point which has associated with it a probability above T2 is assigned a probability of 1 of being in the background. Those pixels which do not exceed T2, but which have less than T1 horizontal or vertical neighbours not exceeding T2, are also assigned a probability of 1. Fig. 3.1.c, 3.2.c and 3.3.c show the results of applying this process to fig 3.1.b, 3.2.b and 3.3.b.

- As a result of the first step some object points might have been identified as background points (light points in dark areas) and some spurious high "o"-probabilities may still be present (dark small clusters in a light neighbourhood). The second part of the algorithm is oriented towards obtaining areas with smooth changes in the associated values of the "o" or "b"-probability. This is done by averaging the probability values of a point with those of its neighbours. The results of this step are shown in fig. 3.1.d, 3.2.d and 3.3.d.

When the algorithm is repeated the input parameters will be varied; specifically the threshold T2 will be lowered since in the new smoothed image object points with

high "b"-probability are very unlikely to be found. A new value of T2 will be about 0.5 less than the previous one.

Fig. 3.1.e, 3.2.e and 3.3.e show the first step applied again and then the contours are drawn on the original picture in fig. 3.1.f, 3.2.f and 3.3.f.

CHAPTER 4
THE MATCHING ALGORITHM

4.1 INTRODUCTION TO THE MATCHING ALGORITHM

In a general manner the matching problem can be defined as follows.

Given a set of target elements T and a set of candidate elements M , we want to identify for each target " t_i " the corresponding match " m_i ". If the targets were completely independent from each other, a matching algorithm would have the characteristics below:

- for each target " t_i " the set M would be scanned in order to determine the best match to assign to " t_i ";
- the matches previously assigned to other targets would not in any way help the search for the correct match.

For the class of problems we deal with, this situation is not the case and the knowledge of the relationships between different targets can be used as a means to judge the consistency of the matching and as a heuristic measure for determining other matches.

Let N be the number of targets which constrain each other and let R be a relation on the set T^*N such that if t_1, \dots, t_N constrain each other then (t_1, \dots, t_N) is in

R and viceversa. Also let Q be a relation on the set $(T \times M)^{**N}$ such that if $((t_1, m_1), \dots, (t_N, m_N))$ is in Q then m_1, \dots, m_N can be matches of t_1, \dots, t_N simultaneously. The problem is now to find for each target " t_i " a match " m_i " such that:

if (t_1, \dots, t_N) is in R , then
 $((t_1, m_1), \dots, (t_N, m_N))$ is in Q .

This problem is called the Consistency Matching Problem [18].

In the case under discussion, the sets T and M are the sets of pixels of two digitized pictures and therefore the number N of targets which constrain each other will equal the number of pixels in the target image, say n , and the number of possible matches for each of them will equal the number of pixels in the candidate image, say m . We will finally end up with m^{**n} different possible matchings and among the consistent ones the most consistent must be chosen. Moreover, since the two pictures we deal with are taken from different points of view, part of the scene which appears in the target image could be occluded in the candidate image and then, for some " t_i ", no correct match would exist. This implies that the "null" match must be added to the set of matches and, since, in practice, $m=n$, the final

number of possible matchings will be $(n+1)**n$.

In the actual solution some assumptions have been made in order to solve this problem.

1. Since the pair of images under consideration will have a small disparity we can say that a possible match for target "ti" must be localized in an area with coordinates of the center equal to the x,y coordinates of "ti" in the target image. A reasonable area is a rectangular window with edges of length equal to 1/10 of the corresponding maximum dimension of the picture. Moreover, if the geometry of the system is given, one of the edges of the window can be reduced: for instance, if the two cameras lie on a horizontal plane, the vertical edge of the window can be reduced.
2. Among the possible matches for target "ti", only the ones which are estimated to be the most probable will be considered. The best 3 matches are a satisfactory approximation. This number is also dependent on the size of the window discussed above: the smaller the window, the smaller the number of possible matches that can be chosen.
3. The consistency check is limited to the matches of targets which have the same y coordinate, that is the relation R discussed above will constrain only those targets

with the same y coordinate. This is justified if the two cameras lie on a horizontal plane. In this case, in fact, the distortion between two images is basically horizontal and when the above consistency is satisfied a much more global consistency is automatically achieved. In case of different geometries the relation R should constrain targets with other linear relationships.

4.2 IMPLEMENTATION OF THE MATCHING ALGORITHM

In the current implementation of the algorithm, the most probable matches of target "ti" are given by the best values of the correlation computed as shown in section 4 of chapter 2. Now, let x_i, y_i and x_j, y_j be the coordinates of the targets "ti" and "tj" and let x'_i, y'_i and x'_j, y'_j be the coordinates of probable matches "mi" and "mj". The consistency criterion we have used to check that "mi" and "mj" can be correct matches for "ti" and "tj" states that the length of the vector:

$$m_j - m_i$$

must be in the range:

$$[(t_j - t_i)/h - 1 ; (t_j - t_i)*h + 1]$$

or in terms of x,y coordinates

$$\text{SQRT}[(x'j-x'i)**2 + (y'j-y'i)**2]$$

must be in the range:

$$[(xj-xi)/h -1; (xj-xi)*h +1]$$

since $yj-yi=0$.

These relations can be justified as follows. Let P1 and P2 be two object points and let p1 and p2 be their projections in a generic image plane. The length of the segment p1p2 is a function of the magnitude of the segment P1P2 in space, of the distance of P1P2 from the image plane and of the relative position of P1P2 with respect to the image plane. If another image plane is given, both the distance and the relative position of P1P2 with respect to the new image plane will vary, therefore the magnitude of the projection p1p2 onto this plane will be different. Now, if we consider the projections of P1P2 onto the image planes of the cameras, we can compute, given the geometry of the system, the maximum expectable ratio of the projections t1t2 and m1m2, call it "h". This maximum ratio can be used as a means to check that, for any pair of targets ti,tj, two possible matches "mi" and "mj" define a segment which is consistent with the distance between "ti" and "tj". Eventually, by adding and

subtracting 1 to the above relations we take care of the fact that we are dealing with discrete digitized pictures in which the projection of each point is recorded with an approximation of half a pixel. The details about the computation of "h", are given in Appendix A.

The algorithm proceeds towards the goal of assigning the correct matches to the targets by analysing, one by one, the rows of the target picture.

1. For each element of the row the most probable matches are determined, that is, in a window in the candidate image centered at the same position of the target in the first image, those points for which the correlation values with the target in question are the highest are recorded.
2. Only the most probable matches are first analysed. Among them, clusters of compatibility are searched. A cluster is defined as a set of compatible matches of k adjacent targets. The minimum value of k depends on the resolution of the image; in our implementation k is fixed to 5 (coarse resolution).
3. The compatibility among such defined clusters is checked; each pair of adjacent clusters is checked to see if every element in one of them is consistent with all the elements in the other. If the compatibility is not satisfied, then the clusters are reduced until this is. If,

because of the reduction, a cluster goes below the minimum allowed k , it is discarded. When all pairs of clusters have been checked, the matches which remain in them are assumed to be good matches and will go into the FINAL matching list.

4. The targets which have not been given matches by the previous steps are now analysed. For each of them the list of the most probable matches is scanned in decreasing order of probability until a match, consistent with the ones which are in the FINAL list, is found, if any such match exists. These matches will go into a temporary (TEMP) list from which the largest number of matches compatible with each other will eventually be extracted and added to the FINAL list. This step is repeated until each target has an associated match or till no more matches can be added to the FINAL list. Naturally, at each iteration, matches which have already been analysed will not be considered again.
5. If a target still does not have an associated match, this is computed by linearly interpolating the ones of the closest targets which have been matched, provided that these targets are not at a distance greater than a given value, say three pixels. The targets remaining without a match are associated with the "null" match.

4.3. EXAMPLE

We shall now illustrate this procedure by means of an example. This shows how matches are assigned to a subset of targets points of a single row. The row in question is the 52nd. row of a 100 by 100 image. The targets go from the 62nd. to the 79th. position on the row. The value of "h" has been fixed to 1.6.

- Table 1 lists for each target under consideration the most probable matches (matrix MATCHES) obtained using correlation.
- Table 2 shows the results of the second step, i.e. the initial clusters which have been identified.
- In Table 3 the clusters have been checked to see if they were compatible with each other. Two elements of the first cluster have been discarded. Table 3 also shows the mutual compatibilities between the two clusters. We can see that by eliminating the matches of targets (52,67) and (52,68) from the first cluster, we have minimized the number of matches to be discarded. The matches which remain in the clusters go into the FINAL list.
- In Table 4 the results of step 4 are shown. The FINAL list has been expanded.
- Eventually three matches are computed by linear interpolation and the final FINAL list is in table 5.

TABLE 1

52 62	52 63	52 64	52 65	52 66	52 67	52 68	52 69	52 70
52 57	52 58	52 59	52 60	52 61	52 62	52 63	52 69	52 67
51 57	51 58	51 59	51 60	51 61	52 63	52 65	53 69	53 67
50 57	50 58	50 59	53 60	53 61	53 62	51 63	52 64	52 65
52 71	52 72	52 73	52 74	52 75	52 76	52 77	52 78	52 79
52 70	52 71	52 73	52 74	52 74	52 75	54 79	54 79	52 79
52 68	54 66	54 75	52 73	54 77	52 76	54 78	52 78	54 80
52 67	52 72	50 78	54 76	52 75	53 75	52 76	53 79	50 80

Matrix MATCHES.

For each target (1st. and 5th. row) the 3 best matches are listed. For each target and for each match the y and x coordinates are shown.

TABLE 2

52 62	52 63	52 64	52 65	52 66	52 67	52 68	52 69	52 70
<u>52 57</u>	<u>52 58</u>	<u>52 59</u>	<u>52 60</u>	<u>52 61</u>	<u>52 62</u>	<u>52 63</u>	52 69	52 67
51 57	51 58	51 59	51 60	51 61	52 63	52 65	53 69	53 67
50 57	50 58	50 59	53 60	53 61	53 62	51 63	52 64	52 65
52 71	52 72	52 73	52 74	52 75	52 76	52 77	52 78	52 79
<u>52 70</u>	<u>52 71</u>	<u>52 73</u>	<u>52 74</u>	<u>52 74</u>	<u>52 75</u>	54 79	54 79	52 79
52 68	54 66	54 75	52 73	54 77	52 76	54 78	52 78	54 80
52 67	52 72	50 78	54 76	52 75	53 75	52 76	53 79	50 80

Matrix MATCHES.

Two clusters have been identified by step 2.

TABLE 3

52 62	52 63	52 64	52 65	52 66	52 67	52 68	52 69	52 70
52 57	52 58	52 59	52 60	52 61	52 62	52 63	52 69	52 67
51 57	51 58	51 59	51 60	51 61	52 63	52 65	53 69	53 67
50 57	50 58	50 59	53 60	53 61	53 62	51 63	52 64	52 65
52 71	52 72	52 73	52 74	52 75	52 76	52 77	52 78	52 79
52 70	52 71	52 73	52 74	52 74	52 75	54 79	54 79	52 79
52 68	54 66	54 75	52 73	54 77	52 76	54 78	52 78	54 80
52 67	52 72	50 78	54 76	52 75	53 75	52 76	53 79	50 80

Matrix MATCHES.

The clusters of TABLE 2 after they have been reduced by the consistency check.

	62	63	64	65	66	67	68
71	T	T	T	T	T	F	F
72	T	T	T	T	T	T	F
73	T	T	T	T	T	F	F
74	T	T	T	T	T	T	F
75	T	T	T	T	T	T	T
76	T	T	T	T	T	T	T

Table of compatibilities between the matches in the clusters. The x-coordinates of the corresponding targets are listed along the vertical and horizontal axis.

52 57	52 58	52 59	52 60	52 61	-----
52 70	52 71	52 73	52 74	52 74	52 75 -----

FINAL list.

The matches inserted into the FINAL list after step 3.

TABLE 4

52 62	52 63	52 64	52 65	52 66	52 67	52 68	52 69	52 70
52 57	52 58	52 59	52 60	52 61	52 62	52 63	52 69	52 67
51 57	51 58	51 59	51 60	51 61	52 63	52 65	53 69	53 67
50 57	50 58	50 59	53 60	53 61	53 62	51 63	52 64	52 65
52 71	52 72	52 73	52 74	52 75	52 76	52 77	52 78	52 79
52 70	52 71	52 73	52 74	52 74	52 75	54 79	54 79	52 79
52 68	54 66	54 75	52 73	54 77	52 76	54 78	52 78	54 80
52 67	52 72	50 78	54 76	52 75	53 75	52 76	53 79	50 80

Matrix MATCHES.

The table shows the matches consistent with the clusters.

52 63	52 65	52 76	52 78	52 79
-------	-------	-------	-------	-------

TEMP list.

The matches consistent with clusters are stored in a temporary list.

52 57	52 58	52 59	52 60	52 61	52 63	52 65	-----
52 70	52 71	52 73	52 74	52 74	52 75	52 76	52 78 52 79

FINAL list.

The FINAL list is updated with the largest number of consistent matches from the TEMP list. In this example all the matches from the TEMP list are copied into the FINAL list and then step 4 is not repeated again.

TABLE 5

52 57	52 58	52 59	52 60	52 61	52 63	52 65	52 66	52 68
52 70	52 71	52 73	52 74	52 74	52 75	52 76	52 78	52 79

FINAL list

The FINAL list after linear interpolation.

4.4. MORE DETAILS ABOUT THE ALGORITHM AND ITS APPLICATION TO STEREO PAIRS OF IMAGES

We shall now describe more details about the algorithm we have discussed and then present some examples of its application to stereo pairs of images.

We have said that the matching algorithm in order to define the correspondence between targets and candidates first defines, among the most probable matches, clusters of compatible matches. These clusters are assumed to be a good starting point for global matching, since the consistency among highly probable matches reinforces the probability that each match is correct. Now, in order to ensure the consistency between different clusters, we check the clusters with each other and then reduce them until the consistency is satisfied. In particular, since the evidence that a target has a wrong match comes from the matches associated with its neighbours, we consider only those matches which come from adjacent clusters and which are associated with points in the first image displaced by not more than 10 pixels.

We now assume that the majority of the cluster elements are correct matches for the corresponding targets. This implies that an algorithm which reduces the

clusters has to preserve the largest possible set of compatible matches. In order to do this each element is assigned a degree of incompatibility equal to the number of elements in the other cluster with which it is inconsistent. The element with the highest degree of incompatibility is eliminated first and then the degrees of incompatibility of those elements with which it was inconsistent are reduced by 1. The algorithm is applied iteratively until no more incompatibilities are present. With reference to the example of section 4.3, first the match of target 68 and then the match of target 67 are eliminated.

We now note that in the cluster merging process (step 4) we have to solve a problem similar to the checking of clusters. In other words we have to determine the largest set of elements in the TEMP list which are compatible with each other. The above algorithm is still applicable if we consider the sets of elements which have to be checked against each other as coinciding with the TEMP list.

Fig. 4.1 4.2 4.3, a and b, show three pairs of images to which the matching algorithm has been applied. The first two pairs have been obtained by rotating the box

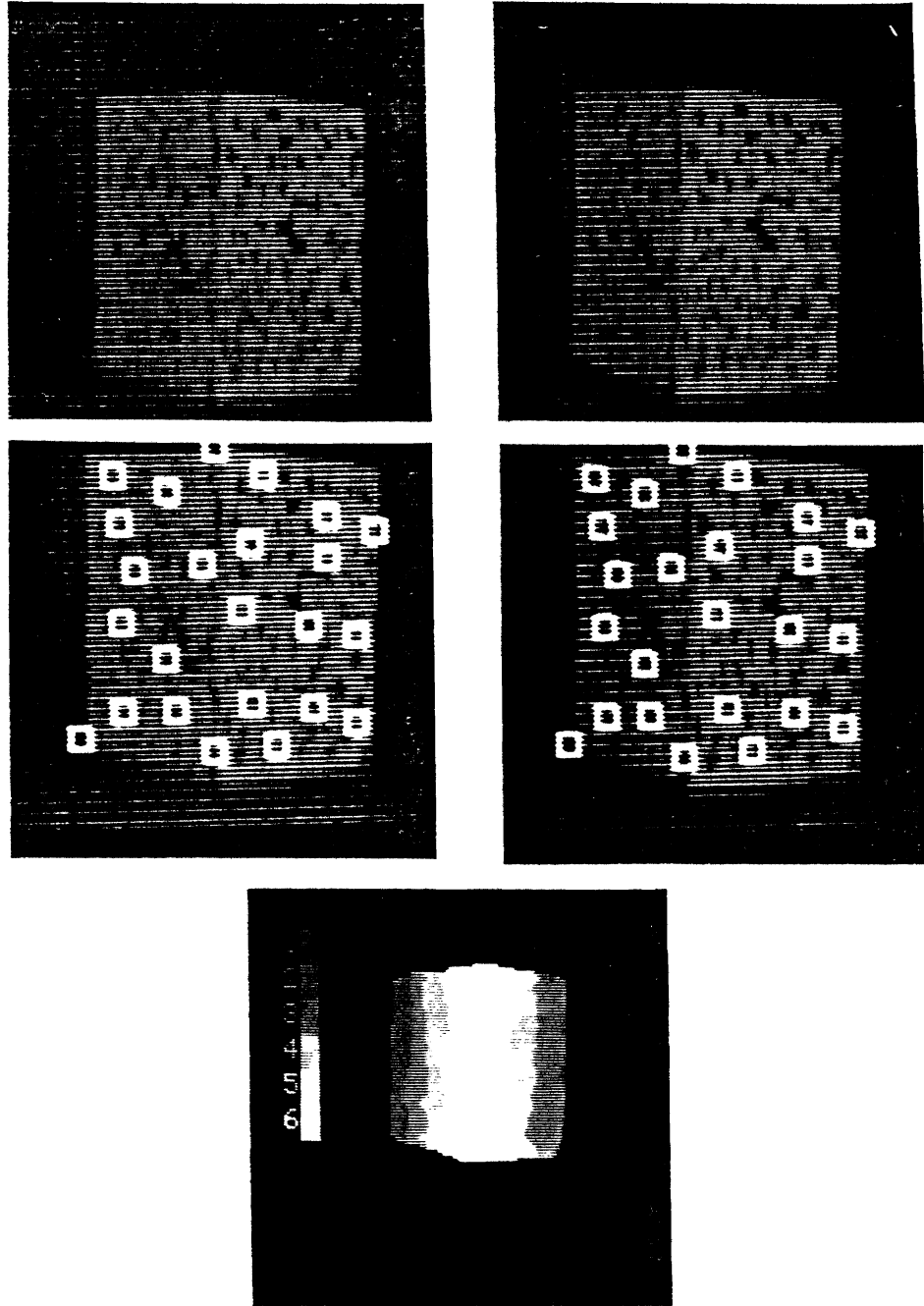


fig. 4.1
MATCHING.

a and b. a stereo pair of views of the box;
d and e. some samples of the matches determined by the
algorithm;
f. disparity map.

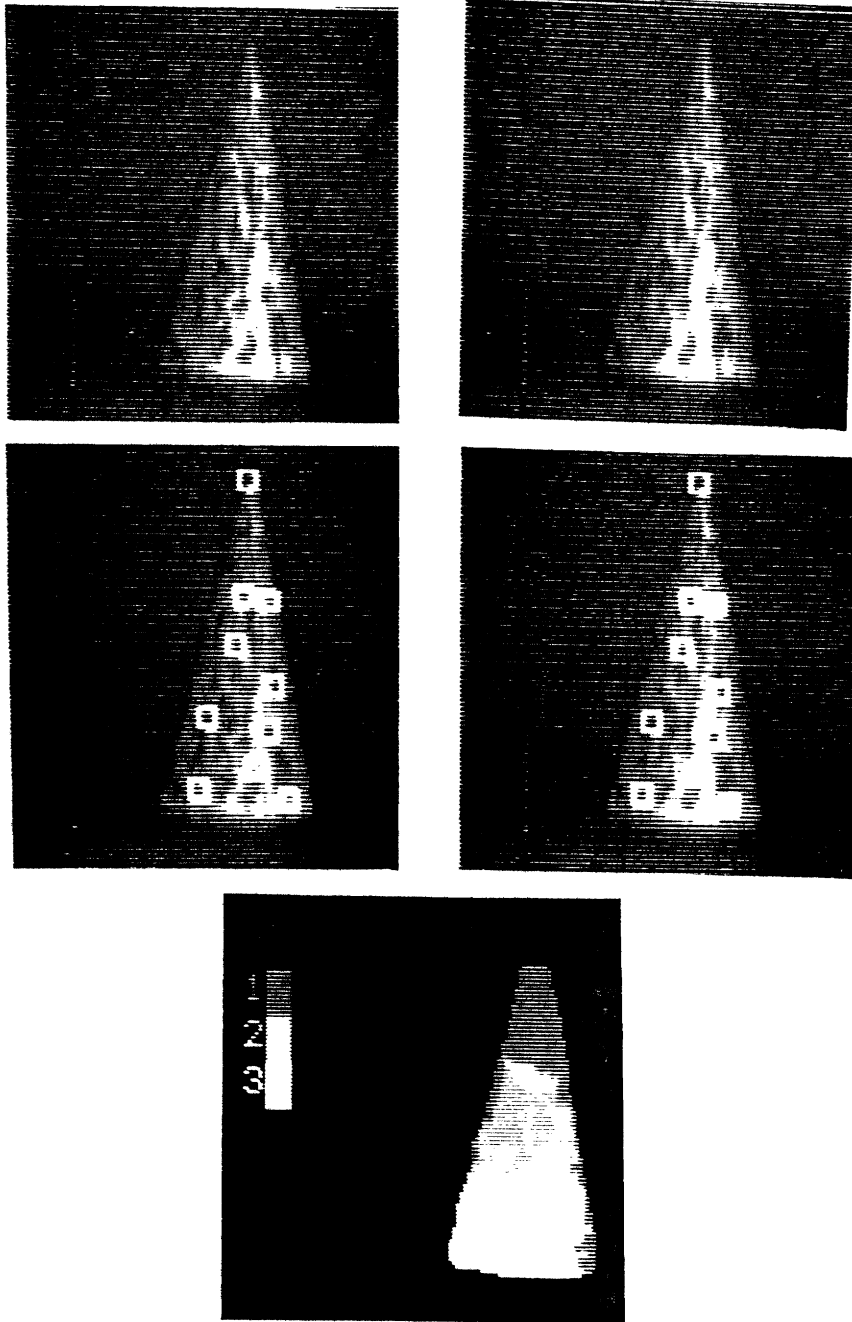


fig. 4.2
MATCHING.

a and b. a stereo pair of views of the cone;
d and e. some samples of the matches determined by the
algorithm;
f. disparity map.

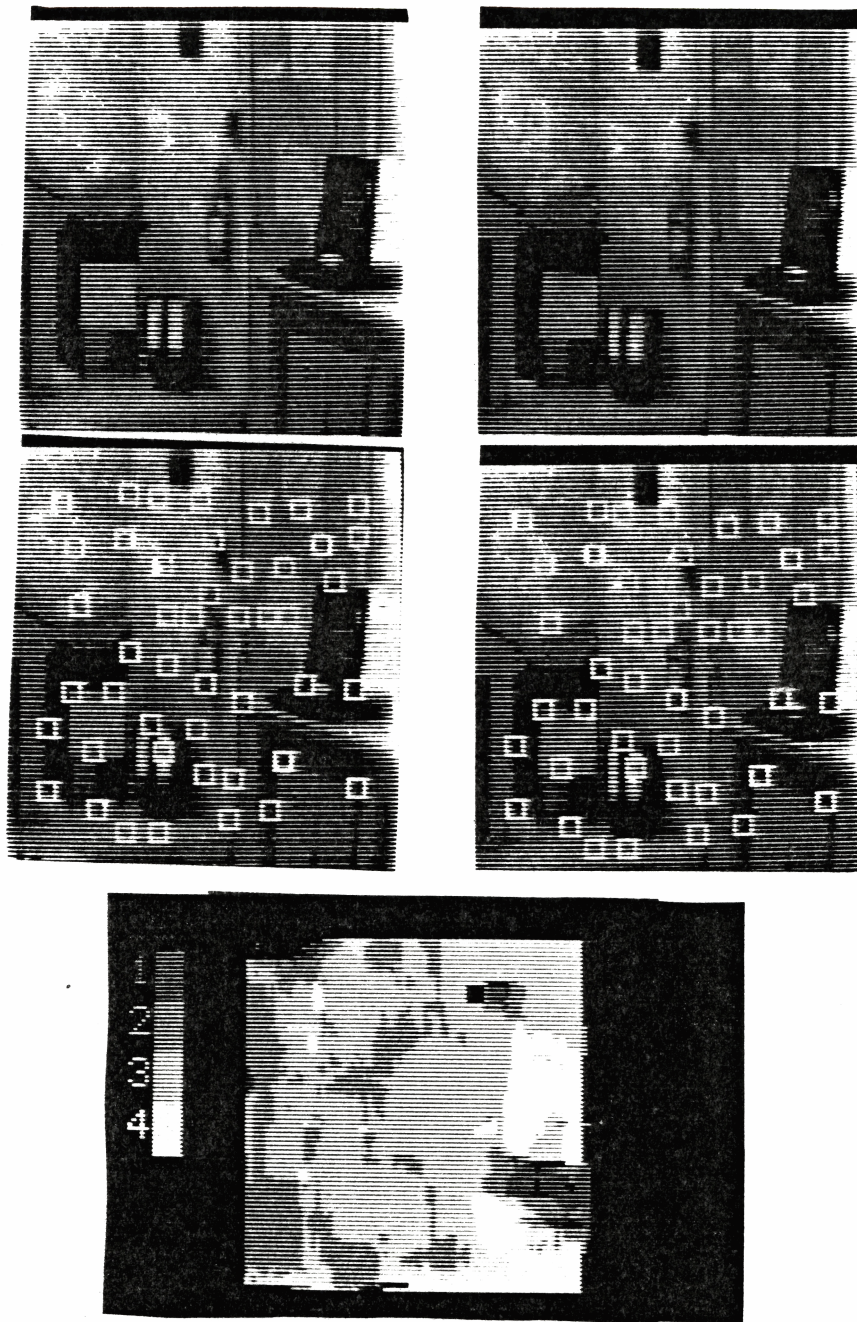


fig. 4.3
MATCHING.

a and b. a stereo pair of views of the Lab;
d and e. some samples of the matches determined by the
algorithm;
f. disparity map.

and the cone by 5 degrees around a vertical axis. This has been chosen so as to be coincident with the hidden vertical edge in the case of the box, and so that it crosses the base circumference in the case of the cone. The pictures of the Lab have been taken from two different viewpoints with the orientations of the camera differing by about 1 degree. The values of "h" used in the three examples are 1.55, 1.55 and 1.24 respectively. Fig. 4.1 4.2 4.3, c and d, show some sample targets and their corresponding matches. Finally fig. 4.1.e, 4.2.e and 4.3.e show the disparity maps computed by the algorithm. They represent the absolute values of the horizontal disparities. In particular, black points correspond to targets which have not been given a match by the algorithm. In the example of the cone note that since the axis of rotation does not pass through the vertex, non zero disparities for points close to the vertex are justified. Note that while the disparity maps of the box and of the cone give a clear idea of the objects under analysis, the map of a more complicated scene does not do so (fig 4.3.e). This is due to the small disparities and to the coarse resolution of the images and also due to the one pixel imprecision of the algorithm (see app.A). In any case it is interesting to observe that the algorithm defines areas of uniform disparity in accordance with the continuity assumption.

APPENDIX A
CHECKING FACTOR

Let P_1 and P_2 be two object points and p_1 and p_2 their projections on an image plane (fig. A.1). On the plane defined by P_1 , P_2 and the center of projection C let A be the angle that the vector P_1P_2 forms with the image plane and let B be the angle that the ray projecting P_1 forms with the perpendicular to p_1p_2 , say " r ". We want to derive an expression for the ratio, say " h ", of the projection of P_1P_2 in two different image planes and then determine the maximum value, say h_{max} , that this ratio can reach. Obviously, the two projections will be defined by a pair of targets and by an associated pair of possible matches. Note that in section 4.2 we have used the notation " h " to refer to h_{max} .

First we note that when, in one of the views, P_1 and P_2 lie on the same projecting ray, the corresponding projection $p_1p_2=0$ and h_{max} becomes infinite. Since in this case $A=90-B$, in the following we will assume A to be less than $90-B$. In practice A will be constrained to be less than the above theoretically permissible maximum. This implies some limitations for the algorithm and we will discuss them later.

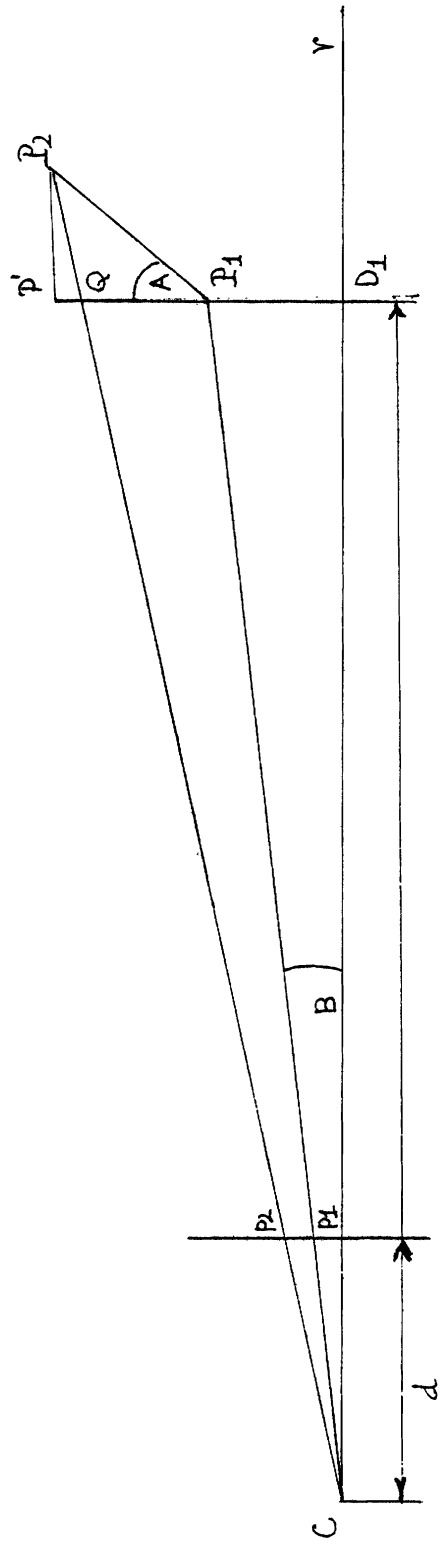


fig. A.1

With reference to fig. A.1 we can say that:

$$p1p2 = (P1P2 \cos A - QP') d/D1$$

and therefore

$$p1p2 = P1P2 \cos A [1 - QP'/(P1P2 \cos A)] d/D1$$

Since

$$QP' = P1P2 \sin A \operatorname{tg} B$$

therefore

$$p1p2 = P1P2 \cos A (1 - \operatorname{tg} A \operatorname{tg} B) d/D1$$

For given P1 and P2 fixed in space, if we let A1 A2, B1 B2, D11 D12 and d1 d2 be the above quantities specialized for a stereo pair of views, we will obtain:

$$h = [\cos A1 (1 - \operatorname{tg} A1 \operatorname{tg} B1) D12 d1] \\ / [\cos A2 (1 - \operatorname{tg} A2 \operatorname{tg} B2) D11 d2]$$

Eventually, if we assume that the distances D1 and d from the two image planes are constant, we can write:

$$h = [\cos A1 (1 - \operatorname{tg} A1 \operatorname{tg} B1)] \\ / [\cos A2 (1 - \operatorname{tg} A2 \operatorname{tg} B2)]$$

In order to generalize the above expression for

all possible vectors P_1P_2 we assume A to be positive in the anticlockwise direction and B to be positive in the clockwise direction. Note that the largest absolute value of B , say B_{max} , is achieved when the ray projecting P_1 crosses one of the edges of the image plane. Moreover, since we deal with small disparities not greater than 10 pixels, the difference between B_1 and B_2 will be very small, about 1 degree. Finally, for almost horizontal vectors P_1P_2 , the difference between A_1 and A_2 is approximately the disparity angle between the image planes of the cameras.

To derive the maximum value of h we express h as $h=h_1+h_2$ where

$$h_1 = \cos A_1 / \cos A_2$$

$$h_2 = (1 - \operatorname{tg} A_1 \operatorname{tg} B_1) / (1 - \operatorname{tg} A_2 \operatorname{tg} B_2).$$

h_2 is clearly maximum when $\operatorname{abs}(A_1)$ is less than $\operatorname{abs}(A_2)$, and $\operatorname{abs}(A_2)$ is the maximum allowed. Under the assumption $\operatorname{abs}(A) < \operatorname{abs}(90-B)$ the function $1 - \operatorname{tg} A \operatorname{tg} B$ ranges between 0 and 2 and shows the maximum slope near the extremes, that is when $\operatorname{abs}(A)$ and $\operatorname{abs}(B)$ are both maximum. The maximum value of h_2 is found when both numerator and denominator go to 0 and not to 2 since, in this case, for the same variations, smaller quantities are being divided. In

other words, the maximum value of h_2 is reached when $\text{abs}(A_1)$ and $\text{abs}(B_1)$ are less than $\text{abs}(A_2)$ and $\text{abs}(B_2)$ respectively and when A_2 and B_2 hold the maximum values and have the same sign. Since h_1 and h_2 are maximum for the same values of A_1 and A_2 , h_{max} is reached when:

$$A_2 = \text{abs}(A_{\text{max}})$$

$$A_1 = \text{abs}(A_{\text{max}}) - (\text{angle between the image planes})$$

$$B_2 = \text{abs}(B_{\text{max}})$$

$$B_1 = \text{abs}(B_{\text{max}}) - \sim 1$$

Now, to allow h_{max} to be a general check for the matching procedure we cannot choose it either too large or too small. A large value of h_{max} would not be a good check for small variations of the projections in the two image planes, while a small value of h_{max} would consider to be wrong those variations corresponding to large values of A 's and B 's. Moreover, in deciding the value of h_{max} we consider the fact that a wrong match is more frequent when large variations occur. Thus, if h_{max} is not a strict check for small variations, this is not crucial. If we assume for example that the angles between the image planes and the value of B_{max} , which are both constant of the system, are about 5 and 10 degrees respectively, with $h_{\text{max}}=1.54$ we can correctly check situations in which A_{max}

is less than 75 degrees. Note that a large value of A corresponds to sharp variations in the depth of P_1 and P_2 . In this case the continuity assumption is not satisfied.

To conclude we want to stress three points. First, since the targets for which we check the consistency of the matches are in a limited range, fixed to 10 in the implementation, even when the check is generous it does not allow in any case wide variations in the number of pixels. Secondly, to allow a match to be accepted, it must be consistent with the matches of all the targets within a distance of 10. Thirdly, because of the limitation on A , some correct matches could be discarded and some targets remain unmatched. The final linear interpolation can recover some of these matches.

The checking procedure could be improved by defining a correspondence between plp_2 and the associated B angle. In this way, knowing the projections in two different image planes and then B_1 and B_2 , we could compute the maximum possible variation h_{max} as a function of B_1 and B_2 with a consequently better precision for the entire checking procedure. This has not been implemented in the actual version of the algorithm.

BIBLIOGRAPHY

1. H. Baker, "Edge-based stereo correlation", Proc. Image Understanding Workshop, April 1980, pp.168-175.
2. S.T. Barnard and M.A. Fischler, "Computational stereo from an IU perspective", Proc. Image Understanding Workshop, April 1981, pp. 157-167.
3. S.T. Barnard and W.B. Thompson, "Disparity analysis of images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.Pami-2, No.4, July 1980, pp. 333-340.
4. A.J. Danker and A. Rosenfeld, "Blob detection by relaxation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.Pami-3, No.1, January 1981, pp. 79-92.
5. R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis", Copyright by J. Wiley & Sons, 1973, chapter 10.
6. R.O. Duda and P.E. Hart, "Use of Hough transformation to detect lines and curves in pictures", Communications of the ACM, Vol.15, January 1972, pp. 11-15.

7. M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Technical Note 213, AI Center, SRI International, Menlo Park, California, March 1980.
8. D. Gennery, "Stereo camera calibration", Proc. Image Understanding Workshop, November 1979, pp. 101-107.
9. W.E.L. Grimson, "A computer implementation of a theory of human stereo vision", M.I.T. AI Laboratory, AI Memo 565, January 1980.
10. M.J. Hannah, "Computer matching of areas in stereo images", Ph.D. dissertation, Stanford AI Laboratory, Memo AIM-239, July 1974.
11. A. Herskovitz and T. Binford, "On boundary detection", M.I.T., AI Memo 183, 1970.
12. M. Hueckel, "An operator which locates edges in digital pictures", J. ACM, Vol.18, 1971, pp. 113-125.
13. D. Marr and T. Poggio, "A computational theory of human stereo vision", Proc. R. Soc. Lond. B. 204, 1979, pp. 301-328.
14. A. Martelli, "Edge detection using heuristic search

- methods", Computer Graphics and Image Processing, Vol.1, 1972, pp. 169-182.
15. M.P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover", Ph.D. dissertation, Stanford AI Laboratory, Memo AIM-340, September 1980.
 16. A. Rosenfeld, R. Thomas and Y. Lee, "Edge and curve enhancement in digital pictures", University of Maryland, Technical Report, 1969.
 17. A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis", IEEE Transactions on Computers, Vol.C.20, May 1971, pp. 562-569.
 18. L.G. Shapiro and R.M. Haralick, "Algorithm for inexact matching", Proc. 5th. International Conference on Pattern Recognition, Vol.1, 1980, pp. 202-207.
 19. Y. Yakimovsky and R. Cunningham, "A system for extracting three dimensional measurements from a stereo pair of TV cameras", Computer Graphics and Image Processing, Vol.7, No.2, April 1978, pp. 195-210.