[4] # Binocular Stereo Algorithm Based on the Disparity-Gradient Limit and Using Optimization Theory

S A Lloyd[1]

GEC Research Limited
Hirst Research Centre, Wembley, HA9 7PP, UK

Reprinted, with permission of Butterworth Scientific Ltd, from *Image and Vision Computing,* 1985, **3**, 177-181.

ABSTRACT

A new algorithm for stereo matching is presented, based on the idea of imposing a limit on disparity gradients allowed in the matched image. The matching problem will be expressed as one of maximizing a certain function, subject to constraints. Standard methods from optimization theory may then be used to find a solution.

Keywords: stereo matching, disparity gradient, optimization

Binocular stereo vision is the process by which three-dimensional structure is recovered from a pair of images of a scene taken from slightly different viewpoints. The difference in positions causes relative displacements or disparities of corresponding items in the images, and these disparities enable the depth to be calculated by triangulation.

There are three main stages in any binocular stereo algorithm: detecting and locating features to be matched, matching features, and calculating depths. When the camera geometry is known, the last of these stages is trivial. The method used for the first stage has been described elsewhere[1], so this paper will be concerned with the second stage. It will assume that a set of edge points, together with their orientations, has been extracted from each image.

The algorithm to be presented is based on the idea of imposing a limit on disparity gradients allowed in the matched image[2,3]. The disparity gradient between a pair of matched points is given by the ratio of the change in their disparity to the distance between their locations in the monocular image (Figure 1).

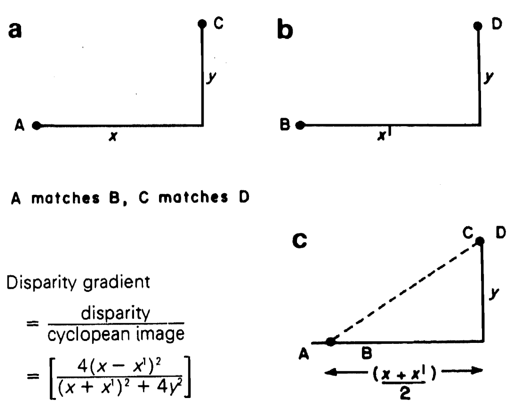

Figure 1. Definition of disparity gradient: a, left image; b, right image; c, cyclopean image

[1] Now with Hewlett Packard Laboratories
Filton Road, Stoke Gifford, Bristol, BS12 6QZ, UK.

It is related to the degree to which the surface on which the two points lie recedes from the observer; or, if the points lie on different surfaces, to the depth between those surfaces. The idea of using a disparity gradient limit is that matches of neighbouring primitives, when considered as a pair, should have a small disparity gradient. Pollard *et al*[2] have developed several algorithms, both iterative and noniterative, which implement this idea, and they have demonstrated their efficiency on a range of natural and synthetic images. These algorithms, however, have little mathematical basis, and the iterative versions have not been proved to converge. The aim of this work was, therefore, to produce an implementation of the idea of disparity-gradient limit based on optimization theory.

CONSTRUCTING A POOL OF POSSIBLE MATCHES

It will be assumed that the cameras are set up so that corresponding edges are constrained to lie in corresponding rows. Edges are then allowed to match provided they have the same contrast sign and similar orientations. The precise definition of similar orientations is that the angles are either within 30° of each other or lie within an 'orientation similarity' limit derived from the disparity gradient limit

$$\frac{4(\tan\theta - \tan\varphi)^2}{(\tan\theta + \tan\varphi)^2 + 4(\tan\theta - \tan\varphi)^2} < L_{dg}^2$$

Where L_{dg} is the disparity gradient limit. This condition is dependent on the edge orientation, the closer an edge is to vertical in one image, the greater the range of potential orientations that can arise in the other image.

OPTIMIZATION APPROACH

To use optimization theory, it is necessary to find a function to be maximized. In other words, a score must be defined for each way of matching the images, so it is possible to say when a match is a good one. Suppose that the images have been matched. For edge i in the left image and edge j in the right image, the decision variable x_{ij} is defined by

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ matches } j \\ 0 & \text{otherwise} \end{cases}$$

For edges i and k in the left image and edges j and l in the right image, a compatibility factor $q(i,j,k,l)$ can be defined in some way to represent the compatibility of matching i to j and k to l. For example q could be defined as

$$q(i,j,k,l) = \begin{cases} 1 & \text{if disparity gradient limit is satisfied} \\ 0 & \text{otherwise} \end{cases}$$

Then a global measure of the goodness of the match is

$$F(x) = \sum_{i,j,k,l} q(i,j,k,l) x_{ij} x_{kl}$$

where $\mathbf{x} = (x_{11}, ..., x_{MN})$, ie the number of pairs of matches, each weighted by its compatibility factor. This is the function that will be maximized.

To apply optimization theory, the definition of the decision variables x_{ij} will be extended to be the probability that edge i matches edge j. Then the uniqueness constraint can be written

$$\sum_j x_{ij} \leq 1 \text{ for all } i$$

$$\sum_i x_{ij} \leq 1 \text{ for all } j$$

The following conditions must also hold

$$x_{ij} \geq 0 \text{ for all } i,j$$
$$x_{ij} = 0 \text{ if } i \text{ is not allowed to match } j$$

The region in MN-dimensional space fefined by the above conditions is called the feasible region. The problem is now to find the point in the feasible region with the largest value of F. If F were convex, this maximum would occur at a vertex of the feasible region. It can be shown[4] that any vertex of this region has the property that $x_{ij} = 0$ or 1 for all i,j, so a decision would be made. Unfortunately, it is not clear whether F is convex or not. In practice, however, this has not been a problem.

GRADIENT METHODS

The idea of a gradient method of solution is to begin with an initial feasible solution and, at each iteration, to determine a direction $\mathbf{r}$ in which to move so that by moving in this direction the value of F is increased, while remaining within the feasible region.

For the feasible region described above, the direction $\mathbf{r}$ must satisfy

$$\sum_j r_{ij} \leq 0 \text{ for all } i \text{ such that } \sum_j x_{ij} = 1$$

$$\sum_j r_{ij} \leq 0 \text{ for all } j \text{ such that } \sum_i x_{ij} = 1$$

$$r_{ij} \geq 0 \text{ for all } i,j, \text{ such that } x_{ij} = 0$$

$$r_{ij} = 0 \text{ if } i \text{ is not allowed to match } j.$$

The 'best' direction is that which maximizes $\nabla F.\mathbf{r}$ subject to $\mathbf{r}.\mathbf{r} = 1$ and the conditions above. This is the direction in which the surface slopes most steeply, but this does not necessarily mean that moving in this direction will yield the greatest increase in F, nor that moving in this direction at each iteration will be the best strategy. It is, however, certainly a good direction in which to move. Unfortunately, finding the best direction is difficult because of the nonlinear constraint $\mathbf{r}.\mathbf{r} = 1$. An aproximation to the best direction can be found by replacing this constraint by $\sum |r_{ij}| = 1$, and this (almost) linear programming problem can be solved by the restricted basis entry simplex method[5].

DESCRIPTION OF ALGORITHM

For ease of notation, several sets will be defined. Let

$$R = \{i \mid \textstyle\sum_j x_{ij} = 1\}$$
$$S = \{j \mid \textstyle\sum_i x_{ij} = 1\}$$
$$Z = \{(i,j) \mid x_{ij} = 0\}$$
$$I = \{(i,j) \mid i \text{ is not allowed to match } j\}.$$

To find $\mathbf{r}$, the direction in which to move, the following optimization problem must be solved at each iteraton: maximize $\nabla F.\mathbf{r}$ subject to

$$\sum_j r_{ij} \leq 0 (i \in R)$$

$$\sum_i r_{ij} \leq 0 \ (j \in S)$$

$$r_{ij} \geq 0((i,j) \in Z)$$

$$r_{ij} = 0((i,j) \in I)$$

$$\sum_{i,j} |r_{ij}| = 1$$

It will be assumed that the compatibility factors q are symmetric, that is $q(i,j.k.l) = q(k,l,i,j)$. Then

$$\nabla F = (Q_{11}, ..., Q_{MN})$$

where

$$Q_{ij} = 2\sum_{k,l} q(i,j,k,l)\, x_{kl}$$

If either $R \neq \{l, \ldots, M\}$ or $S \neq \{l, \ldots, N\}$, then the solution to the problem can be found as follows

Let k,l be such that $k \in R$, $l \in S$ and

$$Q_{kl} = \max_{\substack{i \in R \\ j \in S}} Q_{ij}$$

Let r,t,u be such that $(r,u) \in Z$, $t \in S$ and

$$D_{rtu} = Q_{rt} - Q_{ru} = \max_{\substack{(i,k) \in Z \\ j \in S}} (Q_{ij} - Q_{ik})$$

Let s,v,w be such that $(w,s) \in Z$, $v \in R$ and

$$E_{svw} = Q_{vs} - Q_{ws} \max_{\substack{(k,j) \in Z \\ i \in R}} (Q_{ij} - Q_{kj})$$

Note that, depending on R and S, some of these quantities may not exist. Provided that $R \neq \{1, ..., M\}$ or $S \neq \{1,...,N\}$, however, at least one of these will exist. Then there are three cases.

(i) If $Q_{kl} \geq {}^1/_2 D_{rtu}$, ${}^1/_2 E_{svw}$, then
$r_{kl} = 1$, $r_{ij} = 0$ for all $(i,j) \neq (k,l)$

(ii) If ${}^1/_2 D_{rtu} > Q_{kl}$, ${}^1/_2 E_{svw}$, then
$r_{it} = 1$, $r_{ru} = -1$, $r_{ij} = 0$ otherwise

(iii) If ${}^1/_2 E_{svw} > Q_{kl}$, ${}^1/_2 D_{rtu}$, then
$r_{vs} = 1$, $r_{ws} = -1$, $r_{ij} = 0$ otherwise

If $R = \{1, ..., M\}$ and $S = \{1, ..., N\}$, then the problem is harder. In this case, a possible direction can be found as follows. Let r,t,u,v be such that (r,u), $(r,t) \in Z$ and

$$G_{rtuv} = Q_{rt} - Q_{vt} - Q_{ru} + Q_{vu} = \max_{\substack{(i,l) \in Z \\ (k,j) \in Z}} (Q_{ij} - Q_{il} - Q_{kj} + Q_{kl})$$

Let w,x,y be such that (w,y), $(z,x) \in Z$, $(z,y) \in I$ and

$$H_{wxyz} = Q_{wx} - Q_{wy} - Q_{zx} = \max_{\substack{(i,l) \in Z \\ (k,j) \in Z}} (Q_{ij} - Q_{il} - Q_{kj})$$

There are then two cases

(iv) If ${}^1/_4 G_{rtuv} \geq {}^1/_3 H_{wxyz}$, then
$r_{rt} = r_{vu} = 1$, $r_{ru} = r_{vt} = -1$, $r_{ij} = 0$ otherwise

(v) If ${}^1/_3 H_{wxyz} > {}^1/_4 G_{rtuv}$, then
$r_{wx} = 1$, $r_{wy} = r_{zx} = -1$, $r_{ij} = 0$ otherwise

Having chosen a direction $\mathbf{r}$ in which to move, it is now necessary to find a distance λ such that $(\mathbf{x} + \lambda\mathbf{r})$ belongs to the feasible region and $F(\mathbf{x} + \lambda\mathbf{r})$ is as large as possible. In other words, λ must satisfy

$$\sum_j \left(x_{ij} + \lambda r_{ij}\right) \leq 1 \quad \text{for all } i$$

$$\sum_i \left(x_{ij} + \lambda r_{ij}\right) \leq 1 \quad \text{for all } j$$

$$\left(x_{ij} + \lambda r_{ij}\right) \geq 0 \text{ for all } i,j$$

and must maximize

$$F(x+\lambda r) - F(x) = \lambda^2 \sum r_{ij} r_{kl} q(i,j,k,l) + 2\lambda \sum x_{ij} r_{kl} q(i,j,k,l).$$

In the cases given, simple calculation yields the following values for λ.

(i) $\left(1-\sum_j x_{kj},\ 1-\sum_i x_{il}\right)$

(ii) $\left(x_{ru},\ 1-\sum_i x_{it}\right)$

(iii) $\left(x_{ws},\ 1-\sum_j x_{vj}\right)$

(iv) $(x_{ru},\ x_{vt})$

(v) $\left(x_{wy},\ x_{zx}\right)$

This method, although yielding an approximation to the best direction, is clearly slow and highly nonparallel, since at each iteration no more than four x_{ij} are changed. It is possible, however, to use the same ideas but increase the speed and parallelism by doing the computation given above for each row at the same time. This is possible because the images are assumed to be such that corresponding edges lie in corresponding rows.

CHOOSING AN INITIAL FEASIBLE SOLUTION

Since the function being maximized may be highly nonconvex and have many local maxima, the starting point is important. If the probabilities are set as equal as possible, the initial value of the function should be small, and so the algorithm should have a better chance of finding the global maximum. If the starting point were near a local maximum a gradient algorithm would be more likely to find that local maximum instead of the global one. Initially, the probabilities were all set to $1/N$ where

$$N = \max\left(\max_i \#\{j \mid (i,j) \in I\},\ \max_j \#\{i \mid (i,j) \in I\}\right)$$

Then certainly $\sum_j x_{ij} \leq 1$ for all i and $\sum_i x_{ij} \leq 1$ for all j.

This choice, however, meant that the algorithm spent a long time increasing all the x_{ij} so that the sums were nearer to one, so the starting point now used is

$$x_{ij} = \frac{1}{\max\left(N_i, M_j\right)} \quad \text{where } N_i = \#\{j \mid (i,j) \in I\} \text{ and } M_j = \#\{i \mid (i,j) \in I\}$$

Then, again the inequalities are satisfied. This strategy seems to be successful.

COMPATIBILITY FACTORS

The value of $q(i,j,k,l)$ should fall off as i and k (or j and l) move further apart. It is computationally convenient to define a neighbourhood $N(i)$ round each point i, and define q to be zero if j does not lie in $N(i)$. The general form for q was taken to be

$$q(i,j,k,l) = \begin{cases} \dfrac{p(i,j,k,l)}{1+d(i,j,k,l)} & k \in N(i), k \neq i \\ & l \in N(j), l \neq j \\ 0 & \text{otherwise} \end{cases}$$

where $N(i)$ is the $(2n + 1) \times (2n + 1)$ window centred on i, and $d(i,j,k,l)$ is the square of the average of the distance between i and k and the distance between j and l. Several different definitions for $p(i,j,k,l)$ were investigated.

$$p(i,j,k,l) = \begin{cases} 1 & \text{if } G_d < L_{dg} \\ 0 & \text{otherwise} \end{cases}$$

or

$$p(i,j,k,l) = \begin{cases} 1 - G_d^2/L_{dg} & \text{if } G_d < L_{dg} \\ 0 & \text{otherwise} \end{cases}$$

where G_d is the disparity gradient.

RESULTS

The algorithm was implemented in C on a VAX 11/750 running under Unix, and tested on a number of images both real and synthesized. The first two pairs of images presented here are random-dot stereograms, which enable quantitative evaluation of the algorithm's performance. Each pair depicts three planes: the bottom plane extends over the whole image, while the top two planes occupy a square in the centre of the image. The top plane is transparent, but the middle plane is opaque. The images are presented for crosseyed viewing in Figures 2 and 3 respectively. The disparities of the three planes are 0,1 and 2 in figure 2, and 0,1 and 3 in Figure 3. Thus, in the second case the disparity gradient limit of 0.7 is violated by neighbouring points lying on different planes. The results are summarized in Table 1.

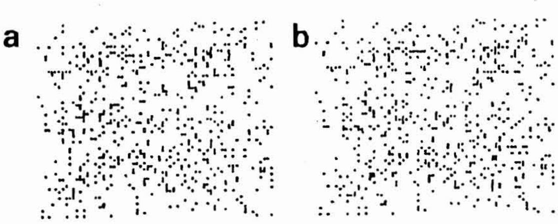

Figure 2. Random-dot stereogram 1: a, right image; b, left image

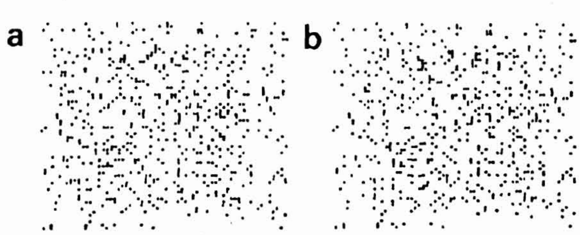

Figure 3. Random-dot stereogram 2: a, right image; b, left image

Table 1. Results of algorithm on random-dot stereograms

Stereogram number	Total number of points	Number matched wrongly	Number left unmatched	% correctly matched
1	618	9	38	92
2	603	56	34	85

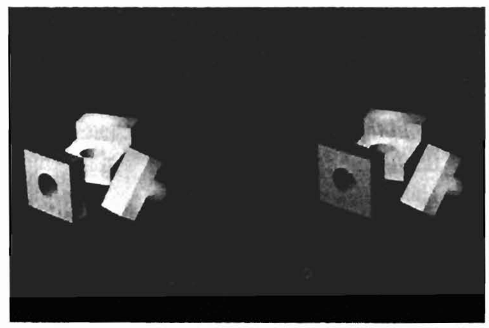

Figure 4. Synthesized pair of images: right image (left); and left image (right)

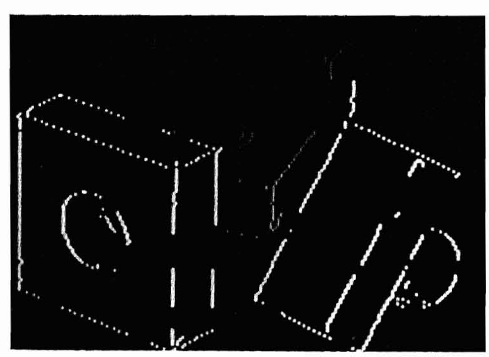

Figure 5. Intensity-coded depth from Figure 4

A synthesised pair of images is shown in Figure 4. The result of the algorithm on this pair is displayed intensity coded (bright is near) in Figure 5 and is displayed as a perspective projection from a different angle in Figure 6. Finally, a pair of real images of a piece of rock is shown in Figure 7, and the result, intensity coded, displayed in Figure 8.

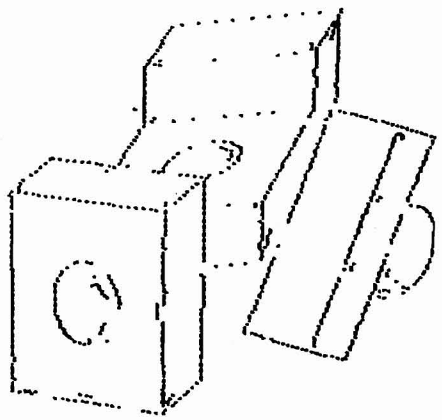

Figure 6. Perspective view of objects in Figure 4

Figure 7. Pair of images of rock: right image (left), left image (right)

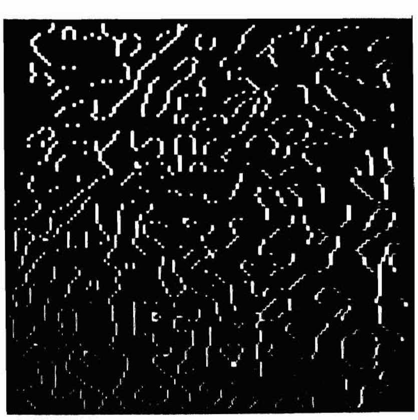

Figure 8. Intensity-coded depth from Figure 7

CONCLUSIONS

The algorithm has been seen to perform well on a range of real and synthesized images, even when the disparity-gradient limit is exceeded.

REFERENCES

1 Lloyd, S.A., (1985) Dynamic programming algorithm for binocular stereo vision. *GEC J. Res.* **3**(1), 18-24.

2 Pollard, S.B., Mayhew, J.E.W. and Frisby, J.P. (1985). PMF: a stereo correspondence algorithm using a disparity gradient limit. *Perception* **14**, 449-470.

3 Trivedi, H.P. and Lloyd, S.A. (1985). The role of disparity gradient in stereo vision. *Perception* **14** 685-690.

4 Garfinkel, R. and Neuhauser, G. (1972) *Integer programming*, Wiley, New York.

5 Hadley, G. (1964) *Nonlinear and dynamic programming*, Addison Wesley, Wokingham UK.