# 308-558B: Fundamentals of Computer Vision Marr & Poggio's Cooperative Stereopsis Algorithm: An Implementation In C

Mani Ghasemlou 9831653

April 9, 2001

# Contents

# 1 Introduction

## 1.1 The problem of stereopsis

Stereo reconstruction can be subdivided into two problems: *the correspondence problem* and *the reconstruction problem.* The task of correspondence is to identify the same features within a stereo pair, and the task of reconstruction is to construct depth information given this correspondence. Generally, the task of reconstruction is fairly trivial: the depth of a feature is a simple function of the solution of the correspondence problem [1].

The heart of the task of correspondence lies in the problem that a feature recognized in one image may be occluded or somehow distorted in the other image. This problem makes the task of correspondence very difficult, if not impossible.

## 1.2 Marr & Poggio's solution

To extract disparity information from a stereo image pair, Marr and Poggio sought to develop a *cooperative algorithm*[2]. Cooperative (parallel) algorithms seem to have greater biological relevance, and it seems that these types of algorithms (neural networks, genetic algorithms, etc) are a popular means of solving problems that seem computationally difficult, but our brains seem to handle with ease.

According to Marr & Poggio, three steps (S) are required to solve the problem of measuring stereo disparity:

- (S1) A feature or location must be selected in the first (left) image.
- (S2) That same feature must be identified in the second (right) image.[3]
- (S3) The disparity of the corresponding image locations is measured.

[1] This is why we won't actually recover feature *depth* in this implementation; we will only recover the disparities between equivalent features in a stereo pair.

[2] By their definition, a cooperative algorithm is a small subset of parallel algorithms that operate on many input elements and reach a global organization by way of local, interactive constraints.

[3] This maps to the "correspondence problem" as discussed in the previous subsection (§1.2).

Marr and Poggio's solution to this problem is based on two constraints (C), which are each mapped to two rules (R) that relate to how the information of an image pair is combined:

- (C1) At a point in time, a point in space has a unique position.
  $\rightarrow$ (R1) Each item from each image may be assigned at most one disparity value.
- (C2) Matter is cohesive, and it is seperated into different objects. The surfaces of objects are generally smooth relative to their distance from the viewer
  $\rightarrow$ (R2) Disparity varies smoothly almost everywhere.

From the above rules, a cooperative algorithm was designed. This algorithm is explained in greater detail within the following section (§2).

# 2 Algorithm Description

## 2.1 Overview of the algorithm

The following equation compactly describes the cooperative algorithm that will be used:

$$C^{(n+1)}_{x,y,d} = \sigma \left\{ \sum_{x',y',d' \in E(x,y,d)} C^{(n)}_{x',y',d'} - \epsilon \sum_{x',y',d' \in I(x,y,d)} C^{(n)}_{x',y',d'} + C^{(0)}_{x,y,d} \right\} \tag{1}$$

where $C^{(n+1)}_{x,y,d}$ represents the state of the point at position $(x, y)$ of disparity $d$ at iteration $n$, $\sigma$ is a simple threshhold function of value either 0 or 1, $E(x, y, d)$ is the "excitatory" elements of the disparity space ("the circular region": the points surrounding an $(x, y)$ position at depth $d$, $\epsilon$ is the "inhibition" constant, and $I(x, y, d)$ represents the "inhibitory" elements ("the thick cable region": the lines going through the disparity space for the different disparities of point $(x, y)$.

The initial state of the solution space[4], $C^{(0)}_{x,y,d}$ is the state containing all possible solutions given an image pair, that is, $C^{(0)}_{x,y,d} = 1$ if $l(x, y) = r(x +$

[4] I will be using "disparity space" and "solution space" interchangeably.

$d, y)$, and $C^{(0)}_{x,y,d} = 0$ otherwise, where $left(x, y)$ and $right(x, y)$ are the values of the left and right images at coordinates $(x, y)$, and $d$ is a possible disparity value.

## 2.2 How it works

This algorithm is basically weighing two elements against each other. In an application of (R2), we add up all the local fusions at a certain depth, and weigh it against the sum of all other fusions for the point in question at different depths. Applying this algorithm through multiple iterations, areas of equal depths will grow and merge, while areas of depth change will be maintained.

Note that while the initial space contains all possible correspondences, after the first iteration there is only one correspondence for each point[5].

# 3 Implementation In C

Once the algorithm (1) was properly understood, implementation in C was fairly straightforward.

The name of the application that applies the cooperative algorithm is called `stereo`. The standard input is as follows:

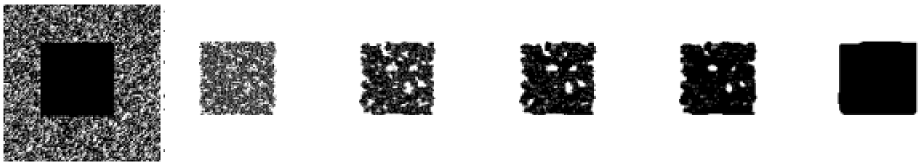
```
stereo <left image> <right image> <output depthmap>
       <iterations> <epsilon>
```

where the first two parameters are the left and right images (in `pbm` format, the third is the output file (in `pgm` format, where the intensity represents disparity), the fourth is the number of iterations to carry out, and the fifth is the value for $\epsilon$ in (1).

## 3.1 Overview of objects used

**loadImage.c** This contains functions for loading `pbm` (portable bitmap) image files. It does not support the full `pbm` specification, most importantly pertaining to comments within the files.

[5]This is a relfection of (R1).



(a) (b) (c) (d) (e) (f)

Figure 1: The algorithm applied on the trivial case. (a) initial disparity space (b) one iteration (c) two iterations (d) three iterations (e) four iterations (f) ten iterations

**writeImage.c** This contains functions for writing out disparity maps as `pgm` (portable graymap) files. The lighter the intensity, the "higher" it is.

**marrPoggio.c** This is the implementation of the Marr and Poggio algorithm.

# 4 Test Cases

Please refer to the included image files for the actual left/right stereo pairs. The images tend to lose their integrity when being scaled and printed, making it impossible for even the human brain to solve the correspondence problem. Therefore they were not included with this document (to the naked eye they simply appear as two noisy squares).

**A trivial example**

This tests a random-dot stereo pair that gives the impression of a square having been "cut out" or "pushed in" from the image plane. The solution obtained after approximately 10 iterations was acceptable. See Fig.1 for the different iterations, and Fig.2 for the resulting depth map.

**3-D checkerboard**

Here, we test the algorithm on the more complicated case of a 3-D checkerboard, where alternating squares seem to be pushed in and pushed out. This example took longer to converge to a good solution than the other ones. See

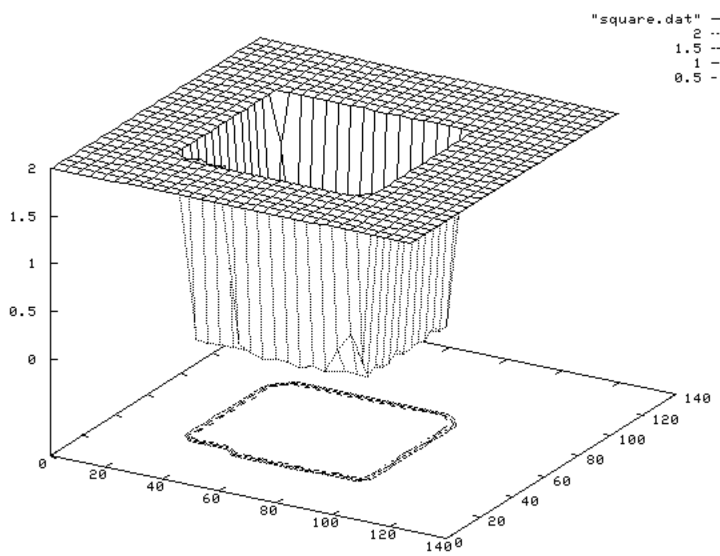



Figure 2: Depth map of a trivial case

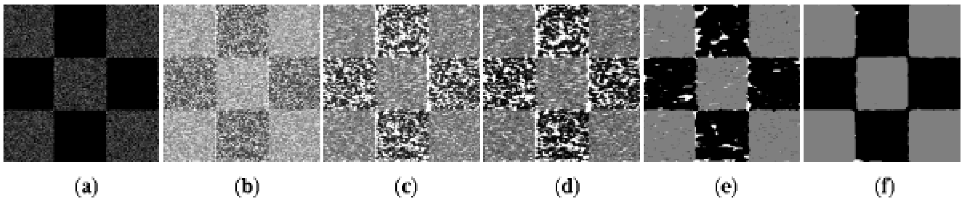



Figure 3: The algorithm applied on the checkerboard case. (a) initial disparity space (b) one iteration (c) two iterations (d) three iterations (e) ten iterations (f) twenty iterations

Fig.3, Fig.4 for the depth map after 10 iterations, and Fig.5 for the (more satisfactory) depth map after 20 iterations.

**A happy example**

Now, we'll test the algorithm on a stereo pair that contains a large number of curves. Note the early convergence. See Fig.6 for the different iterations of the algorithm, and Fig.7 for the resulting depth map after 10 iterations.

**Julesz's stereo pair**

This stereo pair was obtained from Dr. Bela Julesz's home page. See Fig.8, and Fig.9 for the resulting depth map. Dr. Bela Julesz is a very important figure in the realm of depth perception; consult the references for more information about him.

**The number two**

This is a difficult example for the algorithm, because it contains many different depths which vary in steep gradients. See Fig.10 for the different iterations, and Fig.11 for the resulting depth map.

# 5 Conclusion

Marr & Poggio's algorithm is important in confirming the theory that human depth perception is not entirely based on object recognition. The random-

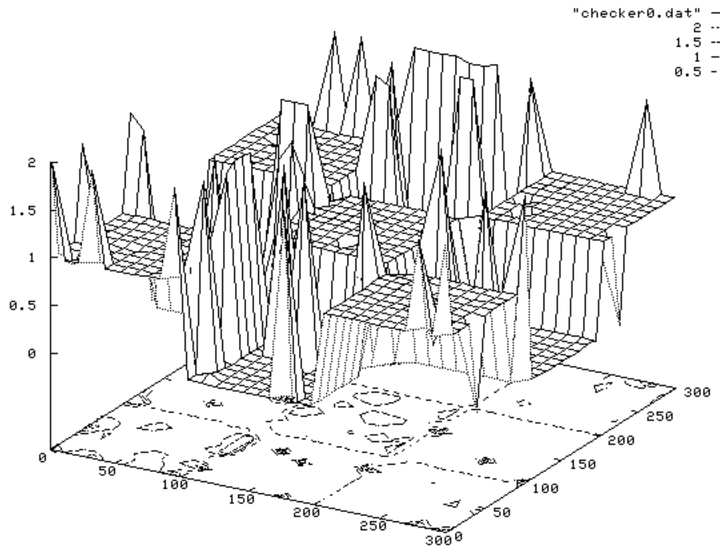



Figure 4: Depth map of a 3-D checker board, after 10 iterations. Note that, while the actual resulting image for this iteration seemed acceptable, the depth map reveals the importance of having the appropriate depth planes merge to a single level.

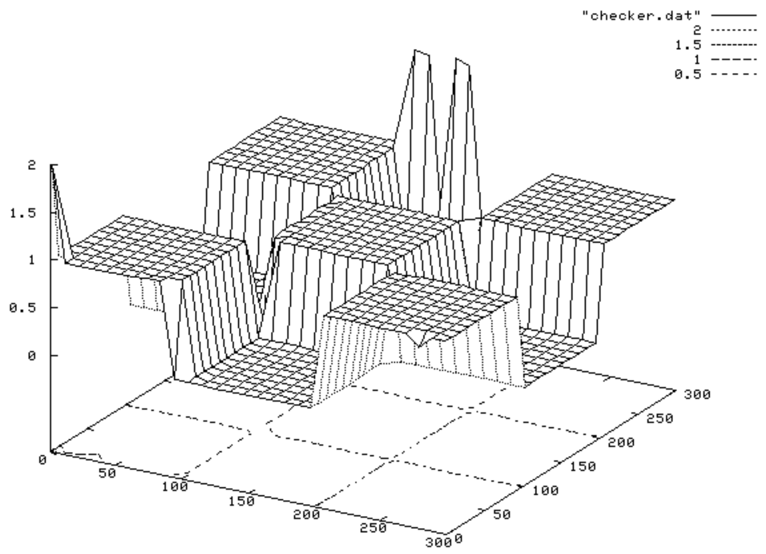



Figure 5: Depth map of a 3-D checker board, after 20 iterations

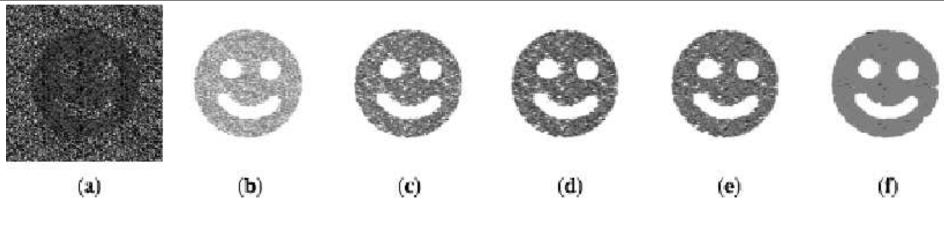



Figure 6: The algorithm applied on the smiley example. (a) initial disparity space (b) one iteration (c) two iterations (d) three iterations (e) four iterations (f) ten iterations

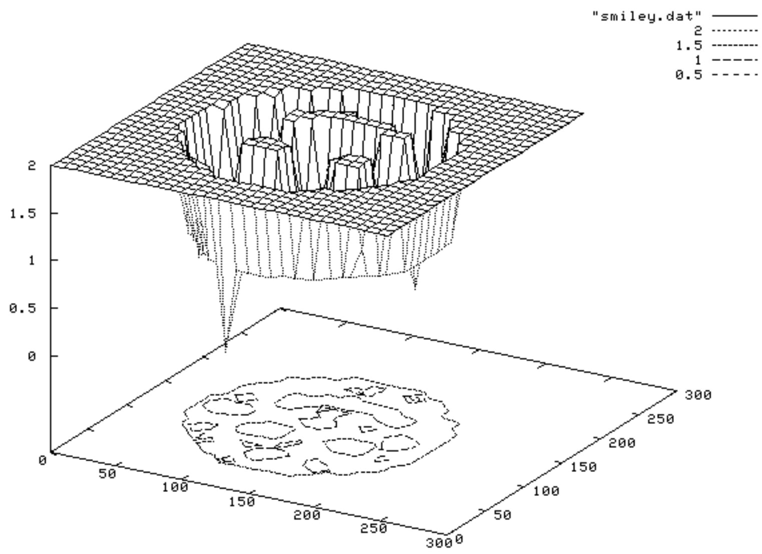



Figure 7: Depth map of the smiley example, after 10 iterations

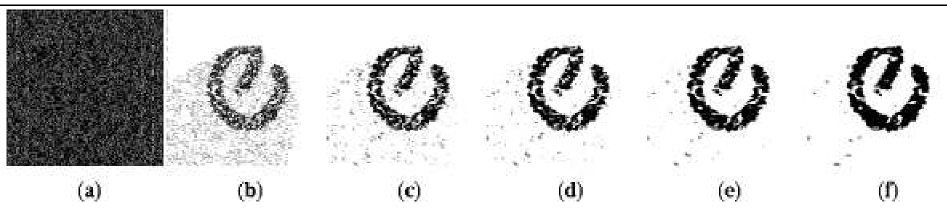



Figure 8: The algorithm applied on the Julesz stereo pair. (a) initial disparity space (b) one iteration (c) two iterations (d) three iterations (e) four iterations (f) ten iterations

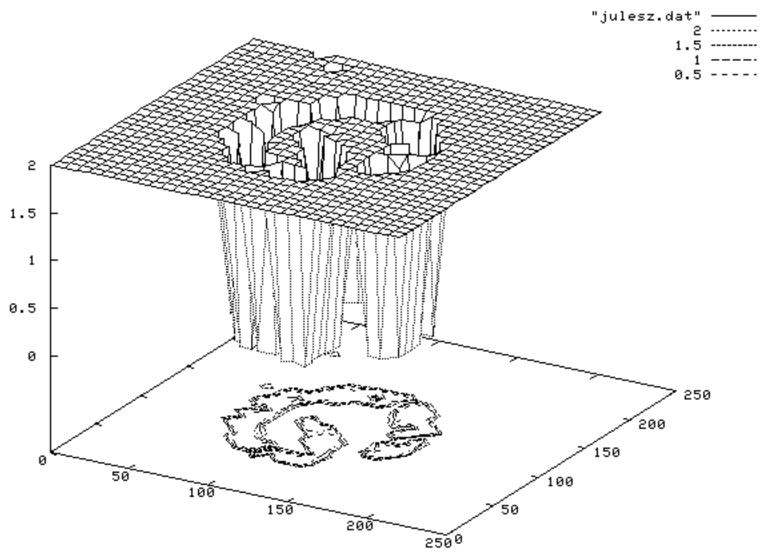



Figure 9: Depth map of Julesz's random dot stereogram, after 10 iterations

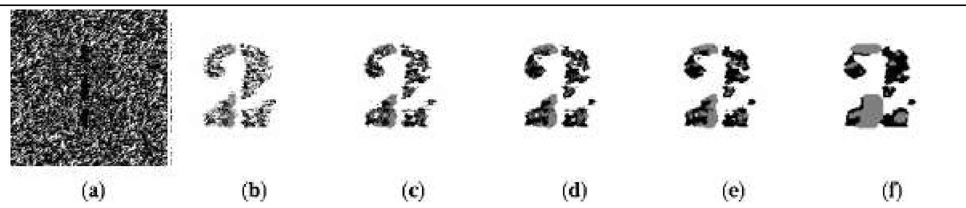



Figure 10: The algorithm applied on the "two" example. (a) initial disparity space (b) one iteration (c) two iterations (d) three iterations (e) four iterations (f) ten iterations

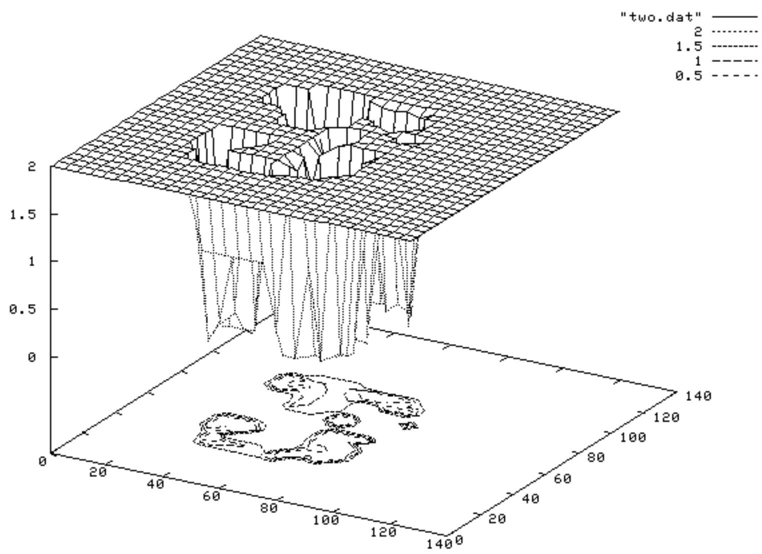



Figure 11: Depth map of the difficult "two" example, after 10 iterations

dot stereogram, and our ability to recognize the depth encoded within them, is proof that our brains apply some kind of correspondence algorithm even when concrete objects are not available to provide proper correlation.

This algorithm succeeds in solving the correspondence problem for random-dot stereo pairs. When applied to these types of images, it converges to an acceptable solution within 5 to 10 iterations.

This algorithm fails in finding correspondences within image pairs that have many different depths. Using more than 3 depth levels in the implementation yielded unacceptable results. Applying this algorithm to actual stereo pairs of real life images does not yield acceptable results. This reaffirms the general belief that stereopsis is only a small subset of our depth perception abilities; our other visual processing abilities do much more work in perceiving depth - which is why it is said that the only reason we have two eyes is in case we lose one of them!

# References

[1] D.Marr, T.Poggio, "A Theory of Human Stereo Vision". Massachusetts Institute of Technology A.I. Memo No. 451 (November 1977).

[2] D.Marr, T.Poggio, "Cooperative Computation of Stereo Disparity". *Science*, New Series, Volume 194, Issue 4262 (Oct. 15, 1976), 283-287.

[3] D.Marr, T.Poggio, "A Computational Theory of Human Stereo Vision", Proceedings of Royal Society, London (1979).

[4] E.Trucco, A.Verri, "Introductory Techniques for 3-D Computer vision". Prentice-Hall, New Jersey (1998).

[5] J.P.Frisby, "Seeing". Oxfort University Press, Oxford (1980).