LOCAL CONSTRAINT INTEGRATION IN A CONNECTIONIST MODEL OF STEREO VISION

by

Charles V. Stewart

and

Charles R. Dyer

Computer Sciences Technical Report #726

November 1987

Local Constraint Integration in a Connectionist Model of Stereo Vision

Charles V. Stewart

Charles R. Dyer

Computer Sciences Department
University of Wisconsin
Madison, WI 53706
608-262-6613

Abstract

The stereo matching algorithms proposed in the research literature have been fairly successful, but none has completely solved the problem. We present a number of steps toward improving matching by: (1) analyzing and reformulating a number of important matching constraints, including uniqueness, coarse-to-fine multiresolution, fine-to-coarse multiresolution, detailed match, figural continuity and the disparity gradient. (2) Building an algorithm that integrates the influence of these constraints *cooperatively* and *in parallel* using the General Support Principle. This principle states that except for uniqueness, *only positive* constraint influences are employed. (3) Implementing this General Support Algorithm using a connectionist network. Such a network allows the constraints to be integrated naturally in a parallel, relaxation computation. The resulting connectionist network has been simulated and tested. It generally produces a high percentage (95-99%) of correct matching decisions. This testing has enabled us to understand the types of problems that *locally* defined constraints can not accommodate, and leads to ideas for non-local mechanisms to address these problems.

The support of the National Science Foundation under Grant No. DCR-8520870 and NASA under Grant No. NAGW-975 is gratefully acknowledged.

Thanks also to Professor Narendra Ahuja at the University of Illinois for supplying many of the images.

Table of Contents

1. Introduction

An important goal of intermediate-level computer vision is to obtain three-dimensional information about a scene. Stereo vision, the matching of images obtained from cameras with slightly different views of the world, plays an important role in this process. This matching converts the two-dimensional information available in each image into a three-dimensional depth map. However, the process of determining this matching is itself difficult and computationally intensive. For each point in an image there is a large number of possible matching points in the other image. Usually, at most one of these matches is valid. The goal of the matching process is to select the valid matches and eliminate the invalid ones.

Many solutions to the stereo matching problem have been proposed in the computational vision literature. These tend to be fairly successful, but they also have a number of problems. First, the algorithms tend to produce spurious matches at certain important locations in images, primarily at significant occluding boundaries. Second, they are sensitive to noise in the images. This includes problems with extraneous edges and with missing edges, especially along contours. More significant structural differences between the images cause even greater problems. Third, they often have difficulty producing correct matches when image features include repetitive patterns. Finally, the algorithms are generally slow. The quality of matching might be improved by using more information in the matching process. Unfortunately, with a sequential model of computation, the use of this extra information makes the efficiency problem worse. This may force rich sources of information to be ignored in favor of computationally feasible solutions to the matching problem.

This paper presents a number steps toward solving these problems. First, we present a careful analysis of the constraints used by stereo algorithms to select valid matches. Our analysis exposes weaknesses in the assumptions motivating the constraints, in their definitions, and in their actual use. This results in a new formulation of some of the constraints. It also leads to the conclusion that none of the constraints proposed thus far is completely sufficient for stereo matching. One way to overcome the difficulties with the individual constraints is to integrate their influences. If done properly, many of the weaknesses of the individual constraints can be overcome. This occurs when the different constraints are not susceptible to the same problems.

The need for integrating multiple constraints leads to our proposal of the General Support Algorithm for stereo matching. The algorithm organizes the influence of the constraints using the General Support Principle.

The principle states that, with only one exception, all the constraints used in stereo matching provide *only positive influence* in selecting the valid matches. The constraints incorporated into this algorithm are all defined locally as pairwise interactions between matches. The algorithm is implemented using a *connectionist network*. This provides a natural model for constraint integration by allowing the constraints to interact cooperatively and in parallel to select the valid matches. Connectionist networks structured this way are similar to relaxation networks that have been used frequently in computational vision.[6, 20]

1.1. Stereo Problem

There are several stages to the binocular stereo problem. These include (1) determining the imaging geometry, (2) detecting image features for matching, (3) finding the matches between these features (the *correspondence problem*), and (4) interpreting the results. The focus of our work is the correspondence problem. In it there are many possible matches for each feature (matching primitive). Usually, at most one of these matches is valid. **Constraints** are used to help select the valid matches from among the candidates. These constraints are derived from assumptions about the continuity of surfaces as well as assumptions about the imaging geometry and image formation process.

The standard imaging geometry for the stereo problem is shown in Figure 1. The cameras are assumed to have point lenses and the images are formed through perspective projection. The lines of sight of the cameras are normal to the center of the image plane, they intersect the focal point, and are parallel to the x-axis in the $x-z$ plane. This model of imaging geometry, called the *parallel model*, is common to most of the stereo matching algorithms that appear in the literature.

A number of definitions that are important for the remainder of the paper are described as follows. Given a point (x_0, y_0) in one image, the **epipolar scanline** for (x_0, y_0) is the line in the other image containing candidate match points for (x_0, y_0). In the parallel model the epipolar scanline is *parallel* to the x-axis for each point. This causes some problems when the matching primitives have an orientation component. Those parallel to the epipolar scanline are often ambiguous. Many algorithms ignore the matching of such points (horizontal edges in the parallel imaging model). In the General Support Algorithm we only match a horizontal edge when it is relatively isolated from other horizontal edges.

The **baseline** is the distance along the x-axis between the focal points of the two cameras. In Figure 1 this distance is $2d$.

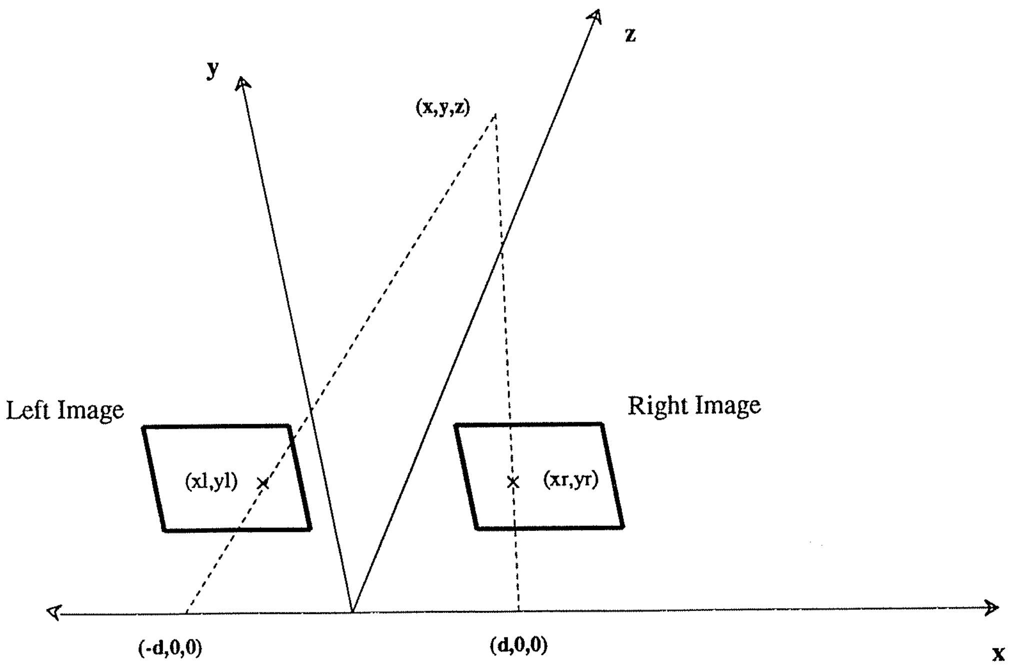


Figure 1. Parallel model of stereo imaging geometry.

The **disparity** of a match is the distance between the matching points in the image. It is used to determine the depth of the scene feature corresponding to that point. If (x_l, y_0) is the left image point and (x_r, y_0) is the right image point, then the distance to the scene point is given by

$$z = \frac{2f\,d}{(x_l - x_r)}$$

where f is the focal length of the lens, and d is as above. Note that points closer to the images involve larger disparities.

Connectionist models are motivated by the perceived organization of the brain.[9,35] In a connectionist network there are simple computing elements (called nodes) and a massive interconnection network among these nodes. The connections communicate the weighted value of the activation of a node. Each node determines its activation based on its prior activation and the input it receives from its incoming connections. The overall computation of the network is organized around iterative changes in the activations of each node. In networks such as the ones used here, the iterations involve feedback between the different nodes. This continues until network settles on a solution. This style of computation is similar to relaxation.[20]

1.2. Previous Work

Stereo matching algorithms can generally be categorized by the types of primitives that they use as input to the matching process. This ranges from point-based matching to matching of sophisticated skeletal structures. A common early approach uses an interest operator to select points in an image and then seeks corresponding points in similars area of the other image.[3, 14] Barnard and Fischler's survey of stereo algorithms highlights a number of these.[4] These algorithms tend to fail when the image data are regular and when there are intensity profile differences between the images, especially at occluding boundaries.

The most common algorithms are those that match edges. Unless they arise from noise, edges may be attributed to physical features of the scene. This makes them more localized than pixels whose values can not be directly attributed to any specific scene feature.[24] The major problem with matching edges is the abundance of candidate matches for each edge. This implies the need for constraints to assist in selecting the valid matches from among the candidates.

To avoid the problem of selecting matches from among numerous candidates, some researchers have attempted to match higher level primitives. These more sophisticated matching structures have fewer candidate matches.[5, 10, 15, 28] This avoids the need for using many constraints. Unfortunately, the structures necessary for matching are difficult to reliably extract. Also, slight appearance differences between the two images often cause significant differences in the structures that are extracted.

In the remainder of this section we briefly discuss a number of important general approaches that have been taken. These include cooperative algorithms, dynamic programming algorithms, and algorithms employing multiple constraints. The analysis of the major constraints used in these algorithms, including uniqueness, multiresolution,[24] figural continuity,[27] and the disparity gradient,[32, 33] is deferred to Section 2.

Cooperative matching techniques were employed by Marr and Poggio in their first algorithm in which they proposed the compatibility, continuity and uniqueness constraints.[26] A number of cooperative matching algorithms have subsequently been proposed.[3, 22, 39, 40] One major problem with cooperative techniques is that they incorporate only a few constraints. Thus, these algorithms (1) do not incorporate additional constraints such as figural continuity and multiresolution that are useful in matching, (2) are unable to detect situations that the constraints can not handle, and (3) require a large number of iterations. However, recently, researchers have attempted to incorporate the search for disparity discontinuities directly into regularization equations.[7] If successful this will remove a major problem in (2).

A second approach incorporates matching constraints into dynamic programming algorithms.[1,2,23,29,34] These algorithms work on each epipolar scanline separately, matching the edges from left to right within the scanline. Thus, they rely heavily on uniqueness and the lack of reversals (the ordering constraint). Dynamic programming techniques have tended to work fairly well, although they employ some restrictive assumptions. It is also not clear how well they adapt to repetitive image structures, noise and significant occlusions (they handle these by allowing edges to go unmatched). Finally, since dynamic programming is essentially a sequential operation, the stereo algorithms relying on it tend to be slow. This latter criticism is somewhat mitigated by the recent development of specialized hardware that handles simple images in a few seconds.[30]

A number of researchers have employed multiple constraints in solving the matching problem. This includes work by Baker and Binford,[2] Eastman and Waxman,[8] and Grimson.[12] Most of these algorithms use the constraints sequentially. Thus, each constraint is used to correct the matching errors of prior stages. Usually, once a correct match is rejected it can not be recovered. The General Support Algorithm presented here offers an alternative. It integrates the constraints cooperatively, and in *parallel.*

1.3. Organization of the Paper

In the remainder of the paper we develop the General Support Algorithm and present results of testing it. Specifically, Section 2 analyzes important matching constraints. This analysis leads to a reformulation of a number of the constraints as well as the conclusion that no individual constraint is sufficient to completely solve the stereo matching problem. Section 3 uses the analysis of the constraints and the ideas involved in connectionist models to develop the General Support Algorithm. The simulation and testing of this algorithm are described in Section 4. Finally, in Section 5 we discuss improvements to the General Support Algorithm using non-local mechanisms to explicitly handle difficult matching situations that locally-defined constraints can only handle implicitly through cooperative interaction.

2. Matching Constraints Analysis

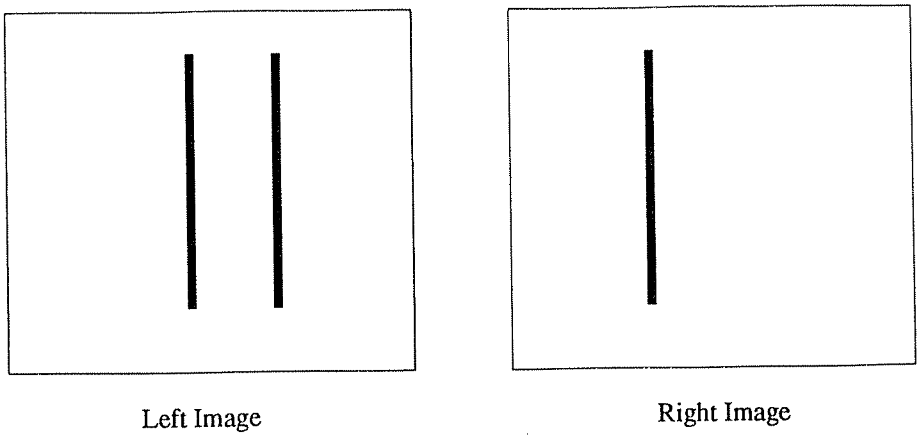
This section presents a careful analysis of the major matching constraints. This analysis emphasizes the assumptions behind the constraints and their general formulations. The main result of this will be the conclusion that no constraint is sufficiently general to work in all matching circumstances. Also, based on our observations, new formulations of some of the constraints are described. In this analysis we assume that the constraints are incorporated into an edge-based matching algorithm. In such an algorithm compatibility predicates are used to determine which edges might match. In our case, a pair of edges, one from each image, form a candidate match if (1) they are in the same image row, (2) their disparity is within a maximum allowed disparity determined by the minimum distance to a point in the scene, and (3) they have compatible orientations.

2.1. Uniqueness

The uniqueness constraint has been used either explicitly or implicitly by nearly every stereo matching algorithm.[26] It is stated as follows: *Each matching primitive in an image should match at most one primitive from the other image.* It is derived from the assumption that each image point corresponds to one point in the world. Uniqueness is the *only* match selection constraint that will be used to *inhibit* candidate matches in our General Support Algorithm. This is allowed because it is a statement about match limitations. All the other constraints make statements about what constitutes a valid match.[24]

The assumption behind the original form of the uniqueness constraint is not always valid when the matching primitives are edges. Specifically, the smoothing process involved in edge detection may cause more than one feature to be merged into a single edge in one image. Because of the different perspectives of the two cameras, these merged features may appear as distinct edges in the other image. A case where human do not enforce uniqueness is shown in Figure 2.[26] In the figure, the result of matching is the impression of one vertical bar floating above another in space. Thus, the vertical line in the right image matches both bars in the left.

The example shown in Figure 2 motivates our reformulation of the uniqueness constraint. It is based on Marr and Poggio's definition of *one-sided uniqueness*,[24] which states that for a given match between image points l and r, at least one of the two points must have no other match. In the General Support Algorithm we define uniqueness in terms of two separate influences on a candidate match (l,r). The first is from the strongest of the competing matches for l. The second is from the strongest of the competing matches for r. In the context of our algorithm, this will enforce normal uniqueness except in occasional circumstances. In addition, it will be able to gradually force the correct interpretation of some ambiguous matches. Without separating the two uniqueness inputs this would not be possible.



Left Image

Right Image

Figure 2. Non-unique matching bars.

2.2. Detailed Match

The detailed match constraint states that candidate matches where the edges have similar appearance are more likely to be valid. This follows from the assumption that the images are taken from similar viewpoints, so that edges are likely to have a similar appearance in both images. Unfortunately, differences in the cameras, inaccuracies introduced by digitization, and slightly differing viewpoints can cause significant variations in the appearance of an edge in the two different images.

Detailed match complements the compatibility requirements used in defining the candidate matches. In determining compatibility, the two edges must have roughly equivalent orientations. In determining detailed match, we compare the average intensities on either side of the edges involved in a candidate match. This has been used in a number matching algorithms.[1,2,23] In our formulation, two intensity comparison measures are employed. The first measure compares the intensity values on both sides of the edges. The second provides support when the intensities on only one side of the edges match. Matching edges that satisfy single-sided similarity often occur near occlusions where there is a discontinuity in disparity.

2.3. Multiresolution Matching

Correspondence between similar matches at multiple resolutions provides another useful constraint. It is based on a number of assumptions about the relationship between edges detected at different resolutions. It was first proposed by Marr and Poggio [24] implemented by Grimson,[11] and has been used frequently since then.[12,17,38,45,47] Marr and Poggio's definition was strict enough that no other constraints were needed. There are a number of problems with such a use of coarse-to-fine matching. These relate to the assumptions behind multiresolution matching. First, as shown by Mayhew and Frisby,[27] matching by humans can occur outside of the range of the coarsest filter. This implies that there ought to be disparities larger than the range of this filter. Thus, simply choosing the nearest match at the coarsest level, as in the Marr-Poggio algorithm, is not sufficient. Second, some of the edges at coarser levels may be averages of a number of features at finer levels. Matches between such edges will not always provide a meaningful measure of disparity for edges at finer resolutions. Third, there might not be any relation between the edges detected at different resolution levels.[33] For example, in an image with grass behind a picket fence, only the fence will appear at coarser resolutions.

In spite of these problems, multiresolution can be a useful matching heuristic. Our reformulation of it relies on an analysis of multiresolution edge detection. First, in general, edge errors can occur at any resolution.[16] At the coarser resolutions they may come from averaging a number of features. At finer levels errors arise due to noise in the image. Second, edge locations tend to be less accurate at coarser resolutions. However, the position of a particular edge only changes locally between resolution levels.[46] Finally, edges which persist across multiple resolutions tend to be physically significant.[25]

In the General Support Algorithm we use multiresolution matching in two ways. The main use of it is similar to Marr and Poggio's approach. That is, matches at a given resolution are used to provide support for matches at the next finer resolution (i.e. coarse-to-fine). The disparities of the finer level matches must agree with the disparities obtained at the coarser levels, and the finer level edges must be in nearly the same spatial position as the coarse level edges. The second use of multiresolution matching is to allow coarser level matches to receive support from similar matches with similar disparities at the next finer resolution level (i.e. fine-to-coarse). This is an important new use of multiresolution since coarse level matches often receive little support through other constraints. Note that this definition involves pairwise interactions between matches, whereas the Marr-Poggio definition used the coarse results to locally re-align the images for finer resolution matching.

2.4. Figural Continuity

The figural continuity constraint was developed by Mayhew and Frisby as part of a competing theory to those proposed by Marr and Poggio.[27] The constraint states that edges along a contour should match edges along a similar contour in the other image. It is based on two assumptions: (1) contours in the scene appear similarly in both images, and (2) contour extraction and matching can occur cooperatively. This is the

Binocular Raw-Primal Sketch (BRPS) conjecture.

The similar contours assumption of figural continuity is not completely general for a number of reasons: (1) groups of edges arising from fine texture are not always part of smooth contours, (2) occasionally, a single contour in an image can arise from multiple contours in the scene at different depths, (3) figural continuity assumes that scene contours appear similarly in the two images. The latter is not always the case in real images because of noise in the image formation process and because of appearance differences between the two images. In addition to these problems, figural continuity does not provide any means of distinguishing between matches along similar contours. In spite of these problems we will make strong use of figural continuity. Our formulation will be a purely local one where figural continuity is determined pairwise between matches. This avoids some of the above problems, but it is still sensitive to gaps in contours caused by noise.

Figural continuity is one example of the continuity constraints that have been used for stereo vision.[26] Each makes assumptions about object, depth or surface continuity that are usually true, but are violated in certain circumstances. In the next subsection we discuss constraints with similar assumptions about surface continuity, but that provide support between matches for edges that are not locally known to be part of a contour. Together these constraints will provide a strong formulation of object continuity assumptions.

2.5. Disparity Gradient and Gaussian Support

A constraint that has recently received a lot of attention is the disparity gradient.[32,41,42] It is a measure of similar disparity values for two candidate matches. Similar ideas were developed by Prazdny using a Gaussian-support model based on the assumption of object cohesiveness.[33] Both the disparity gradient and Gaussian-support are based on the assumption that objects occupy a well-defined three-dimensional volume. The constraints thus assume that points from the surface of an object appear with relatively similar disparities in an image. The disparity gradient is formally defined as follows. Suppose l_1 and l_2 are points in the left image, and r_1 and r_2 are points in the right image. Further suppose that (l_1, r_1) is a candidate match and (l_2, r_2) is a candidate match. Let p_1 be the point half way between l_1 and r_1, let p_2 be the point half way between l_2 and r_2. Further, let $disp\ (l, r)$ be the disparity of a match. Then the matches support each other if

$$\frac{|\ disp\ (l_1, r_1) - disp\ (l_2, r_2)\ |}{D(p_1, p_2)} \leq 1$$

where $D(p, q)$ is the Euclidean distance between two points. (The value 1 is called the disparity gradient circumstances, or (2) to incorporate a number of the constraints in such a way that the weaknesses of one constraint are compensated for by the strengths of the other constraints. We examine each of these possibilities in turn.

The search for general constraints motivates much of the work in stereo vision. While this research may produce improved constraints, it is unlikely that an all-encompassing constraint will be produced. Such a constraint must be based on general assumptions about objects in the world. In addition, it must account for differences in the appearance of objects in both images, including noise, and variations in the two images arising naturally from occlusions and differing camera positions. These two goals reflect the need for generality in defining the constraint. They conflict with the third requirement - that is, the constraint must be strong enough to accurately discriminate between all of the valid and invalid candidate matches.

The alternative approach to overcoming the weaknesses of the individual constraints involves integrating a number of them. This is the approach taken in the General Support Algorithm. We use the following guidelines in constructing such an algorithm:

- The algorithm should work on a wide range of images. This includes random-dot stereograms and natural images. In addition, domain-specific assumptions should not be incorporated into the algorithm.
- The algorithm should overcome the weaknesses of the individual constraints. This is simply a restatement of the main reason for using multiple constraints. In situations where one constraint is weak or inapplicable, other constraints should provide matching information.
- The algorithm should produce a higher percentage of correct matching decisions (correct matches, or correctly finding no match for an edge) when more constraints are providing useful matching information. Thus, any conflict between the constraints should not show up as a reduction in the number of correct matching decisions.
- The algorithm should work effectively in matching situations that have been problematic for other algorithms. This includes noise that shows up as spurious matches and missing primitives, occlusions, and periodic image features.
- Ideally, the use of additional constraints should not slow down the algorithm, at least in a parallel implementation. This may be possible in a connectionist network, where the cost of using additional information is in the number of nodes and connections defined. Thus, if we assume that such an

implementation is possible, we may coarsely measure the time for the network in terms of the number of iterations required.

3.2. The General Support Algorithm

The design of the General Support Algorithm (GSA) attempts to meet the above guidelines for constraint integration. It employs constraints that are defined locally as pairwise relations between candidate matches, or functions of a single candidate match. The GSA is essentially an iterative, parallel, relaxation algorithm. Using the constraints, each candidate match propagates its strength to other matches, and in turn is influenced by those matches. This is repeated until the GSA determines which matches are valid.

In the remainder of this section we describe the algorithm in detail. This includes the General Support Principle which we use to organize the influences of the constraints, the local definition of each constraint and how the algorithm follows the constraint integration guidelines. The details of its connectionist implementation are given in Section 3.3.

3.2.1. The General Support Principle

The GSA organizes the influence of the constraints using the *General Support Principle (GSP).* This principle states that, except for uniqueness, *constraints are only used to support candidate matches*. There are a number of reasons for defining the GSP and using it to organize the interactions of the constraints. First, however, we consider some of the implications of the GSP as it affects what the GSA will and will not accept as a valid match.

In the absence of noise and occlusion, uniqueness would be sufficient for eliminating all invalid matches since every edge would have at least one valid match. However, in reality there are invalid matches that can not be removed through the uniqueness. To solve this problem we also iteratively *decay* the strength of a candidate match. During each iteration of the GSA, candidate matches receive support from other matches, altering their strengths. After each iteration decay is used to reduce these strengths. Thus, noise matches that receive little support will be suppressed over time. In general, the valid matches are those whose constraint support overcomes the inhibiting influences of uniqueness and decay.

In justifying the General Support Principle the main observation is that the constraints are derived from assumptions about objects and their appearance in images. These assumptions lead to assertions concerning relationships between valid matches. For example, the disparity gradient's assumption is that pairs of matches from the *same surface* will satisfy the disparity requirements. There is no claim about pairs of matches from different surfaces. They will often violate the disparity gradient limit. (Prazdny[33] gives a similar argument.) Similar arguments can be made about the other constraints.

Another justification, following from the local definition of the constraints, can be given in defense of the General Support Principle. Specifically, since the constraints are defined as pairwise relations between candidate matches, there is no way to locally distinguish between the following possibilities: (1) the constraints are outside the assumptions of the constraints, or (2) one or both of the matches are invalid. For example, two matches from different surfaces are allowed to violate the disparity gradient requirements, whereas matches from the same surface are not. With purely local constraint interaction these cases can not be distinguished. We rely on the lack of support for a match (or weaker support than one of its competitors) to eliminate it. This is a "least commitment" strategy in defining the local interactions between the constraints.

3.2.2. Constraints Used in the GSA

The General Support Algorithm incorporates the following constraints: uniqueness, disparity gradient, figural continuity, coarse-to-fine and fine-to-coarse multiresolution support, and detailed match. Except for uniqueness, each of these constraints is defined locally as pairwise relations between candidate matches, or as functions of the two edges involved in a candidate match. The local nature of each constraint is either built into the assumptions behind the constraint or in the actual formulation of it. In the remainder of this section we discuss the influence of each constraint and how some constraints complement each other in their definition of support.

Uniqueness is necessary for any stereo algorithm matching primitives as simple as edges since there are usually multiple matches for each edge. The particular form of uniqueness defined here is useful for resolving matches when there are periodic patterns.

The **disparity gradient** and **figural continuity** are complementary surface structure constraints. Both reflect surface continuity assumptions. Figural continuity provides strong support between matches that locally appear to be along the same contour. The disparity gradient gathers support over a broader region based on similar disparities. It provides support between textural features of a surface and between points on a contour that are too distant for figural continuity's local contour requirements. Figural continuity is sensitive to gaps in contours and cannot discriminate between locally-similar contours. The disparity gradient overcomes these difficulties by gathering support in a larger region. On the other hand, the disparity gradient sometimes provides only weak discrimination between matches. Figural continuity is usually a much more certain indicator that a match is valid. In ambiguous situations, matches receiving strong figural continuity support are strengthened more quickly than other matches. The strengthened matches will then reinforce other matches through the disparity gradient.

Multiresolution is most useful for identifying persistent features of the images and propagating support between appropriate matches for these features at various resolution levels. This accelerates the matching process, and provides strong indication of disparity for the most significant features. This helps to resolve the matches for ambiguous regions at finer resolutions. Note that, contrary to previous algorithms,[24] multiresolution support is computed simultaneously at all resolution levels.

Detailed match serves two main purposes. First, since it provides an additional initial indication of what matches might be valid, it tends to accelerate the matching process. Second, it can help overcome ambiguities due to periodic image patterns. It does this by providing additional similarity measures for edges in a match using the regions surrounding the edges. When a coherent set of matches in a periodic pattern receive this support, the relaxation process of the General Support Algorithm will eventually select them as valid.

One commonly used constraint, the **ordering constraint** (i.e. matches must maintain the order of the edges in each row of the images) is not used in the GSA. There are two main reasons for this. first, it relies on limiting assumptions about the types of surfaces. More importantly, it is not necessary. None of the constraints will allow support between matches that violate the ordering constraint (the disparity gradient between these matches is greater than 1). Thus, in order for two nearby matches violating the ordering constraint to both be accepted as valid, they must both receive strong support *independently* in the same neighborhood. This is exactly when the ordering constraint should not apply. In all other cases the activity of the General Support Algorithm will implicitly implement the ordering constraint.

Finally, because of the General Support Principle, a number of other constraints can easily be incorporated into the matching algorithm. Two examples of this are focal gradient[31] and optical flow measures.[43, 44] They would both add support as a function of the edges of a candidate match. In the case of optical flow, this support is added when the flow vectors of the two edges match, indicating approximately similar depths for the edges. The focal gradient has been used to obtain an approximate measure of depth by measuring the spread in the response of the edge detector. The additional support can come either from (1) comparing the blur measure of the two edges, or (2) using the depth obtained from the focal gradient as a coarse measure of disparity.

3.2.3. Following the Guidelines for Integrating Multiple Constraints

Below we list how the General Support Algorithm follows the guidelines for integrating multiple constraints. Section 4 contains further discussion based on the experimental results.

- The GSA should work on many different types of images. There are two main reasons why this is true: (1) the algorithm matches edges, which can be reliably extracted in a wide range of images, and (2) as we have seen, the GSA employs no domain-dependent assumptions.
- The GSA overcomes the weaknesses of the individual constraints. This can be seen in two ways: (1) the situations in which the constraints are prone to error do not overlap. This is most easily seen in the complementary definitions of the surface continuity constraints: figural continuity and the disparity gradient. (2) Because a number of constraints are used, the individual constraints can be restricted to matching situations where they work best.
- Section 4 emphasizes the quality of the matching by the GSA, but unfortunately, appropriate statistical data from other algorithms are not available.
- The GSA should work in a number of difficult types of matching circumstances. Solutions to some of these problems are reflected in the formulation of the constraints and in their integration. For example, one common problem is that of fragmented contours. Figural continuity is sensitive to this problem, but the disparity gradient helps to overcome it by gathering support in a wider region. Another example is that the use of one-sided uniqueness and detailed match sometimes helps to correctly handle scenes with periodic features. Finally, the problem of finding matches for occluded edges is implicitly addressed by the use of decay and uniqueness. An explicit solution requires non-local matching mechanisms (see Section 5).
- The use of more constraints does not increase the time required by the algorithm. Additional constraints can increase the number of iterations needed for matching because of inconsistencies with other

constraints. However, because of the General Support Principle, this conflict can only occur indirectly through support for competing matches. Also, since we rely on a number of constraints, we have implemented each constraint more conservatively.

3.3. Connectionist Implementation of the GSA

The General Support Algorithm is implemented using a hierarchical connectionist network. The lower levels represent the input images and implement edge detection. The highest level is the matching network. It is designed so that (1) each node represents a distinct potential match, and (2) most of the constraints are implemented directly in the connections between candidate match nodes. The matching works through a sequence of iterations. At each iteration each node computes its new activation and output. This new activation is based on the old activation, the decay rate, the supporting input and uniqueness.

In the remainder of this section we consider the activation and output functions of the nodes in the network, the implementation of the constraints, weight assignments for the constraints, and some additional issues concerning the design of the GSA.

3.3.1. Node Activation and Output Functions

The activation of a node depends on the combined influences of the supporting constraints, uniqueness, decay and the node's prior activation. The supporting input to a match is a weighted, linear combination of the constraint input, given by:

$$I_i = \sum_{j=1}^{N_i} O_j \, w_{ji}$$

where the O_j are outputs from other match nodes, and the w_{ji} are the connection weights. Uniqueness input is given by:

$$B_i = \beta \max(O_j \mid j \in Left_i) + \beta \max(O_k \mid k \in Right_i)$$

where $Left_i$ is the set of competing matches for the left edge of match i, and $Right_i$ is the set of competing matches for the right edge. This implements our formulation of uniqueness. The new activation is:

$$A_i = (1 - delta) A_i + I_i + B_i$$

where δ is the decay factor. The output, O_j is simply a threshold function of the activation:

$$O_i = \begin{cases} A_i, & A_i \geq \phi \\ 0 & A_i < \phi \end{cases}$$

where ϕ is the threshold parameter. The activation is limited to the range [-1..1], and the output is allowed to be in the range [0..1]. When the activation of a node reaches 1.0 it is said to be *saturated*, and it remains at 1.0 for the duration of the matching procedure.

3.3.2. Constraint Implementation

As noted previously, most of the constraints are implemented locally as pairwise relations between candidate matches or functions of individual matches. In this subsection we consider the details of the implementation of these constraints. (The uniqueness constraint is built directly into the activation function for a match node described above.) Our description here will define the range of support of a constraint and the weight of its connections.

Detailed match requires that the smoothed intensities on either side of the edges of a given match be compared separately. This produces three additional inputs to a candidate match: (1) input when the low intensity side of the edges is similar, (2) input when the high intensity side is similar, (3) input when both sides are similar. These are mutually exclusive. For (1) and (2) the weight of this input is $BASE\_ONE\_SIDE$. For (3) the weight is $BASE\_BOTH\_SIDES$. A small subnetwork is required to determine the detailed match support.

The *disparity gradient* implementation is straightforward. Two matches (l_i,r_i) and (l_j,r_j), support each other if they are within the disparity gradient limit as defined in Section 2.4. In addition, their distance, $distance$, is limited to the range $MIN\_DISTANCE$ to $MAX\_DISTANCE$. Support between matches closer than $MIN\_DISTANCE$ is governed by the figural continuity constraint. $MAX\_DISTANCE$ is determined by the maximum allowed disparity.

To determine the connection weight for the disparity gradient, let $disp\_diff = | disp(l_i,r_i) - disp(l_j,r_j) |$. Then the weight is given by:

$$w_{i,j} = \frac{BASE\_DG}{distance} * \frac{c}{(disp\_diff + c)}$$

where $c \geq 1$ is a constant, and $BASE\_DG$ is a parameter controlling the overall strength of the disparity gradient. The inverse distance term reflects the likelihood that the candidate matches are from the same surface.[32] The $c \ / \ (disp\_diff + c)$ term scales the support as the disparity difference between the matches increases. Without it, the disparity gradient only weakly discriminates between candidate matches for an edge. However, this scaling tends to favor surfaces that are parallel to the image plane. (Note that Prazdny's Gaussian support model does this also.[33]) Increasing c reduces the favoring of fronto-parallel surfaces, but decreases ability to discriminate between matches.

Figural continuity is implemented between pairs of candidate matches over a limited distance. A pair of matches will support each other if they may be part of the same contour in each image and if they meet the disparity gradient requirement. The disparity gradient is used to ensure similarity between the contours. The actual support between these matches is weighted inversely by their distance apart. The weight is scaled using the parameter *BASE_FC*:

$$w_{i,j} = \frac{BASE\_FC}{distance}.$$

The implementation of multiresolution support is straightforward. Coarse support for a match is found by searching the next coarser resolution near a given match for a match that has (1) similar positions and orientations, and (2) nearly identical disparities. The weight is given by *BASE_COARSE_TO_FINE*. Fine-to-coarse support has the same requirements and has a weight of *BASE_FINE_TO_COARSE*. Note that these multiresolution weights involve no distance factor. The main reason for this is that the position of an edge at differing resolutions can vary.

3.3.3. Weight Assignment

The remaining problem in defining the network is to establish the values of the various parameters, and thereby set the connection weights. These parameters are associated with the implementation of the constraints and the definition of the activation function, including the uniqueness parameter, β, and the decay parameter, δ. Our approach to parameter assignment involves three steps: (1) develop heuristics for the relative values of the parameters and their approximate values, (2) use simple synthetic images representing difficult matching problems to refine the values, and (3) fine tune the values through experience with random-dot stereograms and real images.

Many connectionist models use learning algorithms for weight modification. Unfortunately, such algorithms are inadequate for this problem. Those that employ supervised learning[36] would require *a priori* identification of all the valid and invalid matches for a large set of images. Those using unsupervised learning

[13,37] require internal feedback from a more complete vision system to adjust the weights. Without such a system, unsupervised weight adjustment would tend to reinforce errors made in determining the valid matches.

In taking an analytical approach to weight assignment, we claim that there is a fairly wide range of useful values for the parameters. The reasons for this are as follows. First, because of the General Support Principle (GSP) the supporting constraints only conflict indirectly through the uniqueness constraint. Second, because of the iterative, cooperative nature of the algorithm, under normal circumstances the valid matches in a region of one image form a coalition of support. Within the coalition there is usually support through a number of constraints. Thus, precisely set values for these weights are not crucial. The implications of these statements are twofold: (1) the most important parameter assignment involves the relative strength of uniqueness and decay versus the supporting constraints, and (2) important test cases occur when there is strong competition between conflicting groups of matches.

3.3.3.1. Heuristics for Weight Assignment

The observation that the valid matches form a coalition of support provides us with our first weight assignment heuristics. Specifically, when there are a number of conflicting matching interpretations for an area of an image, the algorithm must have enough time to identify the valid matches. This implies that the constraint support parameters be relatively low to allow a consensus to emerge. If the parameter values are too large, the algorithm might accept too many matches. This also implies that the value of the uniqueness parameter, β, needs to be relatively large. When there are nearly equivalent competing matches, relatively minor differences in the support for those matches should decide between them. When $2\,\beta \approx 1.0$ this occurs naturally. The decay rate δ is set much smaller since it mostly concerned with eliminating candidate matches that receive little support.

There are a number of other heuristics that allow us to qualitatively relate the supporting constraint parameters, and to relate these to uniqueness and decay. For example, the relative strength of the parameters for figural continuity and the disparity gradient is defined through two observations. First, both constraints are derived from assumptions about the appearance of an object surface in the images and they complement each other in their definition of support for a match. This implies that $BASE\_FC \approx BASE\_DG$. Second, the disparity gradient defines support over a two-dimensional area (although the matches are relatively sparse), whereas figural continuity is defined over a limited range in one dimension. This implies that

$BASE\_FC \gg BASE\_DG$. The relative weights for these parameters is a trade-off between the two observations. In practice we have found that $BASE\_FC \approx 2 \times BASE\_DG$ works well.

Next, we need to define the multiresolution parameters. Coarse-to-fine is the strongest of the support parameters since it is the least ambiguous. However, it should not be large enough that it can not be overridden by a coalition of support from other constraints. Fine-to-coarse support is used to compensate for weak support that coarser matches receive through other constraints. Thus, we want $BASE\_FINE\_TO\_COARSE \approx \delta$.

To summarize, in addition to some quantitative relations between the parameters we have the following ordering of the parameters:

$$\beta > BASE\_COARSE\_TO\_FINE > BASE\_FC > BASE\_FINE\_TO\_COARSE \approx D > BASE\_DG$$

Finally, the two detailed match weight parameters should be significantly smaller than the others. Clearly, $BASE\_BOTH\_SIDES > BASE\_ONE\_SIDE$. Also, $BASE\_ONE\_SIDE > BASE\_BOTH\_SIDES / 2$ since the appearance of one side of an occluding edge can vary significantly between two images.

3.3.3.2. Further Parameter Tuning Using Synthetic Images

The second step in parameter tuning is to use synthetic images that represent significant tests for the values and the algorithm. One example is the non-unique matching bars of Figure 2. In this example the strength of figural continuity, multiresolution, and slight disparity gradient support along each contour should be enough to barely (this ensures that one-sided uniqueness is usually enforced) overcome one-sided uniqueness and decay. Figure 3 shows another image pair. In this case, assume that a bar in one image could match both bars in the other. Hence, there are two uniqueness inputs for each match. In looking at the figures, it is obvious which bars should match. However, the only difference in support is that the correct matches receive greater disparity gradient support. Thus, with the disparity gradient support between the correct matching bars these matches should be chosen. When the disparity gradient only provides support *along* each bar no matches should be accepted. Other synthetic images include a number of types of periodic patterns. These are useful for refining the detailed match parameters and ensuring that uniqueness can assist in resolving ambiguities.

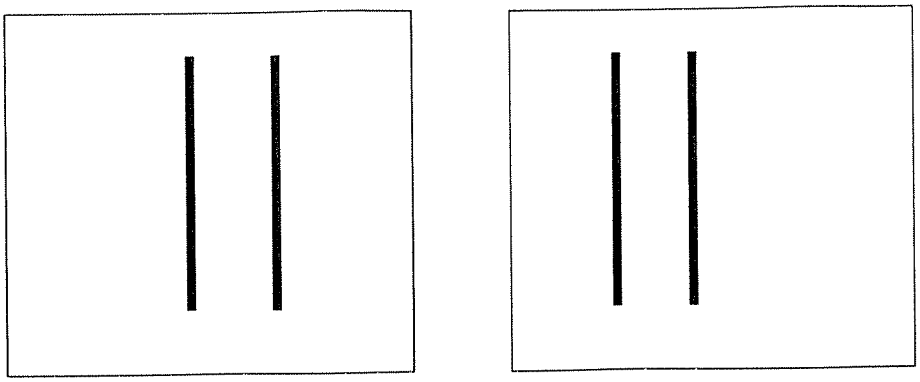


Figure 3. Synthetic image example. See text for discussion.

3.3.4. Further Issues in the Design of the GSA

Before examining the performance of the algorithm, we briefly discuss some additional issues in the design of the General Support Algorithm. These include the size of the network, edge detection, and network termination conditions.

3.3.4.1. Network Size

One of the main problems with the General Support Algorithm is the size of the network. A straightforward implementation of the network described above would require $k\, p\, m\, N^2$ match nodes for a given resolution. Here k is the number of different compatible orientations for a given orientation, m is the number distinct orientations that the edge detector recognizes, p is the number of distinct disparity values, and the image size is N by N. This assumes that there must be a distinct node for every possible match.

An alternative implementation, requiring slightly more complicated candidate match determination, significantly reduces the number of nodes required. Assuming that there is at most one edge at each position, then for any two points, one in each image, there can be *at most* one candidate match. If one node represented all the matches between edges at the two points, a significant savings would be realized. Specifically, the total number of match nodes would be reduced to $p\, N^2$. To realize this savings, two problems must be addressed. First, there must be a means to ensure that only pairs of compatible edges can activate a match node. Second, match nodes no longer represent any orientation information. Figural continuity and multiresolution require this for defining connections, but the disparity gradient, which defines the most connections, does not. The solutions to these problems require the use of conjunctive connections.[9] That is, groups of connections are multiplied together before forming the weighted input to a node. Thus, if the output from the edge detector is binary and there is one conjunctive connection for each compatible set of orientations, only compatible edges will activate a candidate match. Similarly, conjunctive connections may be used to limit figural continuity and multiresolution to compatible orientations. With this additional complexity in defining some of the network connections a significant saving in the number of match nodes and connections is realized.

3.3.4.2. Edge Detection

Another issue we have not addressed completely is edge detection in the lower levels of the network. The General Support Algorithm assumes that the edges are detected and provided as input to the initial phase of matching. In practice we have not attempted the connectionist implementation of an edge detector. This has simplified timing considerations since we assume that all edge inputs are available in the initial phase of matching. The only assumptions that the General Support Algorithm makes about edge detection is that (1) the edges are relatively sparse, and (2) when an edge appears at multiple resolutions, it does so with similar positions and orientations.

3.3.4.3. Termination Conditions

A number of researchers have attempted proofs that their networks will reach steady state conditions.[19] We have no such proof for the General Support Algorithm. There are several reasons for this: (1) the output function is real-valued, making it more difficult to determine what constitutes a steady-state. (2) The structure of the network is irregular. Networks that are known to reach a steady-state are highly regular. (3) The GSA is designed to handle realistic input. Its behavior on arbitrary, uncorrelated data is unpredictable.

4. Experimental Results

This section presents experimental results using random-dot stereograms and real images. The section is divided into four parts. Section 4.1 describes the simulation of the networks. Section 4.2 presents the matching results for random-dot stereograms. Section 4.3 covers results in matching real images. Section 4.4 analyzes the results in terms of the guidelines stated in Section 3.1 for an algorithm that integrates multiple constraints.

In addition to the images presented here we also ran the algorithm on the synthetic images described previously. In each case the algorithm works as desired.

4.1. Connectionist Simulation of the General Support Algorithm

The first step in testing the GSA is designing a realistic, practical simulation of it. The main observation used in doing this is that matches are sparsely distributed. Thus, for a given pair of input images, we build only those nodes that represent a candidate match, and we only define connections between these candidate matches. The result is a subnetwork that is *functionally isomorphic* to the entire network for this pair of images.

The simulation subnetwork is built using the results of edge detection. Edges are located using the Marr-Hildreth edge operator.[25] This operator is shaped with a central positive region surrounded by an extended negative area. In the General Support Algorithm we use three different sizes of this operator, with central positive region diameters $w = 3, 6, 12$. Although not required in theory, in practice a small threshold is used on the magnitude of the zero-crossing, and some minor thinning of non-horizontal edges is necessary. The results of edge detection are provided as input to the simulation. The simulation uses this input to determine the candidate matches and build the connections between them. (No matches are allowed for horizontal edges along contours of length two or more in a given row in an image.) The initial activation for each match is set at a base level plus any additional support from a detailed match. The simulation is allowed to run for either a given number of iterations (usually a max of $\approx$ 16) or until only a small percentage (usually $\approx 1\%$) of the matches have outputs in the intermediate range 0.25 to 0.75.

Finally, the simulation was run using the following parameter values: $BASE\_Coarse\_to\_Fine = 0.225$, $BASE\_Fine\_to\_Coarse = 0.1$, $BASE\_FC = 0.15$, $BASE\_DG = 0.08$, $D = 0.12$, $\beta = 0.5$, $BASE\_Both\_Sides = 0.02$, and $BASE\_One\_Side = 0.12$. All of these values except $BASE\_DG$ were used for all of the test described below. $BASE\_DG$ varies with the density of the matches. This is discussed further below.

4.2. Random-Dot Stereograms

Random-dot stereograms have frequently been used for testing matching algorithms.[21] They are useful for studying matching for two main reasons: (1) there are no monocular depth cues that might assist in the stereo process, and (2) the correct disparity is known for every point. The latter gives an objective measure of matching accuracy that is not available for real images.

Our method of testing stereograms is as follows. All stereograms used are a series of planar surfaces. The 256 by 256 images are first generated with 50 percent dot density. After this the stereograms are treated as normal inputs to edge detection. The network is run using $BASE\_DG = 0.06$. The results of matching are recorded in two ways. First we generate a "disparity image". That is, for each edge with an accepted match in a given image, the disparity of the match is recorded as an intensity. The second method of data collection is to automatically compare the disparities of the matches to the known correct disparities. A match is counted as correct if its disparity is within ±1 of the surface disparity.

An example stereogram is shown in Figure 4. Part (a) of the figure gives the disparity of the surfaces encoded as intensity. Darker points are closer to the viewer. White patches indicate locations that are occluded in the other image. No matches should be found for these points. The background surface is disparity 0; the others are 6, 12 and 18 pixels. Note that in the right image the three non-zero disparity surfaces align. This gives rise to a significant occlusion in the left image. Part (b) of the figure shows the actual stereograms. Part (c) shows the results of intermediate level (level 1) edge detection. Part (d) gives the level 1 disparity image resulting from matching. (The matching ended after 10 iterations.) Points with no edges or no matches appear as white. Table 1 summarizes the statistical results of matching for each resolution level. The second through fifth columns of this table represent, in order, correctly finding no match for an edge, incorrectly finding no match, (0.6 was used as the cut-off between valid and invalid matches) finding the correct match, and finding an incorrect match. The sixth column represents the number of correct matching decisions. A correct decision is either finding the correct match for an edge, or correctly rejecting all matches for an edge. The seventh column gives the number of incorrect decisions. The final set of statistics is given in Figure 5. It shows the percentage of matches whose output is in the intermediate range 0.25 to 0.75 as a function of iteration number. Most matches are determined by the fifth iteration.

	Correct No Match	Incorrect No Match	Correct Match	Incorrect Match	Correct Decision	Incorrect Decision	Percent Correct
Left 0	761	7	16,584	474	17,345	481	97.3
Left 1	442	20	8,087	221	8,308	241	97.2
Left 2	156	39	3,365	289	3,521	328	91.5
Right 0	605	5	16,680	494	17,285	499	97.2
Right 1	409	27	9,278	219	9,278	246	97.1
Right 2	160	25	3,525	289	3,525	314	91.8

Table 1. Matching results for stereogram in Figure 4.

In addition to the high percentage of correct decisions, there are several things to note about the results:

- A large percentage of the errors involve occluded regions. For example, 98 of the 474 incorrect matches at level 0 in the left image are for occluded edges (for 473 occluded edges it correctly found no matches). The numbers are higher in the right image, 217 of 494. In addition, the occluded match errors in one image generally appear as errors for non-occluded edges in the other image. Thus, approximately 315 of 474 errors in the left image involve occlusions. The main reason for the differential between the left and right images in terms of the number of occluded edges with matches is that there is a larger occluded region in the left image. Support between matches in an occluded region is essentially random. More coherent support comes from outside the occluded region. In the left image this support is generally further away from a given match and is therefore weaker. In addition to the matchings for the occluded edges, there are also errors near step changes in disparity that don't directly involve occlusions.
- At the coarsest resolution the results are somewhat poorer. There are two possible explanations for this. First, the edges are somewhat sparser at this level, leading to weaker support for a given match. Second, the errors could simply be due to measurement errors. That is, matches are counted as valid if they are within ±1 pixel of the surface disparity. As the edges become coarser, their position errors tend to increase (see Section 2.3 for our discussion of this), and so do their disparity errors.
- Finally, there are a number of edges for which multiple matches are found. These are counted as errors (161 of them in the left image, level 0) if the two edges in the other image are not adjacent (this can occur when a horizontal contour switches to a more vertical one). The number of double matches is nearly zero at the coarse resolution.

The algorithm was run on several other stereograms as well. The percentage of correct matching decisions for these images was similar to the above example.

4.3. Real Images

The simulations for real images were run in the same manner as stereograms. The matching for each stereo pair was allowed to run for up to 16 iterations or until fewer than 1% of the matches had outputs in the intermediate range. The parameter *BASE_DG* is varied between 0.055 and 0.09 depending on the density of the edges in the images.

Collecting reliable data on the results is more difficult for real images. We attempted to follow similar procedures as for random-dot stereograms in producing disparity images and matching statistics. However, since the ground truth disparity was unknown, we used subjective judgment to determine whether the algorithm made correct matching decisions. The images were graphically displayed, and for each edge, the candidate and accepted matches were highlighted. Statistics on correct decisions were then gathered using the following judgment criteria: First, the disparity of a match was compared to an approximately known disparity for the region of the image. Second, the two images were carefully examined to discern, subjectively, if matching edges appeared to be the same physical edge. This was mostly based on the structure of the images. A limited percentage of the edges were actually checked. This procedure gave a rough quantitative measure of how well the algorithm performs, and allowed us to examine the results in detail to observe where the algorithm succeeded and failed.

We tested the algorithm on the images in Figures 6-14. The figures show the images, intermediate results of edge detection (level 1) and level 1 disparity images. The captions of the figures indicate special features of the results. The statistics for these examples are summarized in Table 2. The second and third column of this table give the number of edges examined and the percent of correct matching decisions. The fourth column shows the parameter BASE_DG used. The fifth through seventh columns of the table give the percent of matches in the intermediate range after the sixth, ninth and twelfth iterations. The final column shows the total number of iterations required. Finally, the graph in Figure 5 shows the number of intermediate nodes as a function of the number of iterations for three tests.

Overall, the results show a high rate of correct decisions. In addition, there are several other general comments to note about the results. First, the disparity images are sparser than the edge images. This is because unmatched edges do not appear in the disparity images. These are usually horizontal edges, but they may also be occlusion edges. Second, the graphs shown in Figure 5 indicate widely varying percentage values and peaks. This variation is due to the percentage of matches that are eventually accepted since most invalid matches never reach the intermediate range. For example, the apple image had 46% acceptance, the rocks image had 19% acceptance, and the sandwich image had 43% acceptance. The main reason for the low value in the sandwich image is that matches are sparser than in the apple image.

The level 0 matching results were not included because some of them were too dense to carefully analyze. Of those that we could analyze, by far the worst performance was for the ball image (89%). In this image the fine resolution edges are extremely dense and regular. This causes a high degree of ambiguity that the algorithm could not completely overcome. We have more to say about this in Section 5.

4.4. Discussion

The General Support Algorithm has been experimentally shown to be successful in producing a large percent of correct matching decisions in the examples tested. Note that we emphasize the percentage of *correct matching decisions*. A correct decision occurs when either the correct match is found for an edge, or when the algorithm correctly finds no matches for an edge. An incorrect decision occurs when an incorrect match is found for an edge, or when the algorithm misses the correct match for an edge. Other algorithms have compared the number of valid matches to the number of invalid matches. Optimizing this figure is misleading since it can lead to an overly sparse set of matches.

In Section 3 we discussed how the GSA was designed following the guidelines for constraint integration. Here, we briefly evaluate its performance in terms of these guidelines. (Unfortunately, it is difficult to separate the influence of the constraints because the constraint definitions used in the algorithm rely on the existence of the other constraints. This makes it inappropriate to compare the performance of the algorithm with and without certain constraints. Also, in examining the results of matching, it is difficult to assign credit or blame to particular parts of the algorithm.)

- The algorithm matches a wide range of images from random-dot stereograms to real images.
- The algorithm overcomes weaknesses in the individual constraints. In observing the results we have seen circumstances where the interaction of the constraints has improved the performance. Most importantly,

	Level 1			% in Mid Range After			
	Number Examined	Percent Correct	BASE_DG	6	9	12	Number of Iterations
Apple	555	98	0.08	2	1		10
Ball	561	98	0.055	4	2	1	13
Books	712	96	0.09	4	2	1	15
Fruit	469	96	0.09	9	4	2	16
Pentagon	565	97	0.055	5	3	2	16
Renault	436	98	0.08	6	3	2	16
Rocks	803	97	0.07	6	3	2	16
Ruts	623	99	0.09	8	3	2	16
Sandwich	493	98	0.09	13	7	5	16

Table 2. Matching Statistics for Real Image Pairs.

in densely textured regions, the matches receiving strong multiresolution and figural continuity support tend to pull the rest of the matches into correspondence through the disparity gradient. Another example of this is where weak figural continuity support was overcome through broader based support given by the disparity gradient. An instance of this is in the Rocks image where the dense fine resolution matches did not have strong figural continuity support between them.

- The GSA produces a high percentage of correct matching decisions. It is unfortunate that there is no way to directly, quantitatively, compare the results with other matching algorithms.
- The GSA works well in circumstances where other algorithms work poorly. It works fairly well in resolving matches for periodic structures and in compensating for minor structural differences between images. In the fruit image (Figure 9), the algorithm chose the matches for the table cloth pattern near the occlusion, but correctly found the matches for this in other places. Also, in general as the structural differences between the two images increases, the quality of the matching decreases. Finally, the GSA handles occlusions fairly well. As was shown by random-dot stereograms, most edges that appear in one image, but are occluded in the other are not matched.
- Finally, even with the use of a number of constraints, the matching did not require very many iterations. This was seen in the percentage of intermediate matches in the middle range as a function of the number of iterations.

5. Concluding Remarks

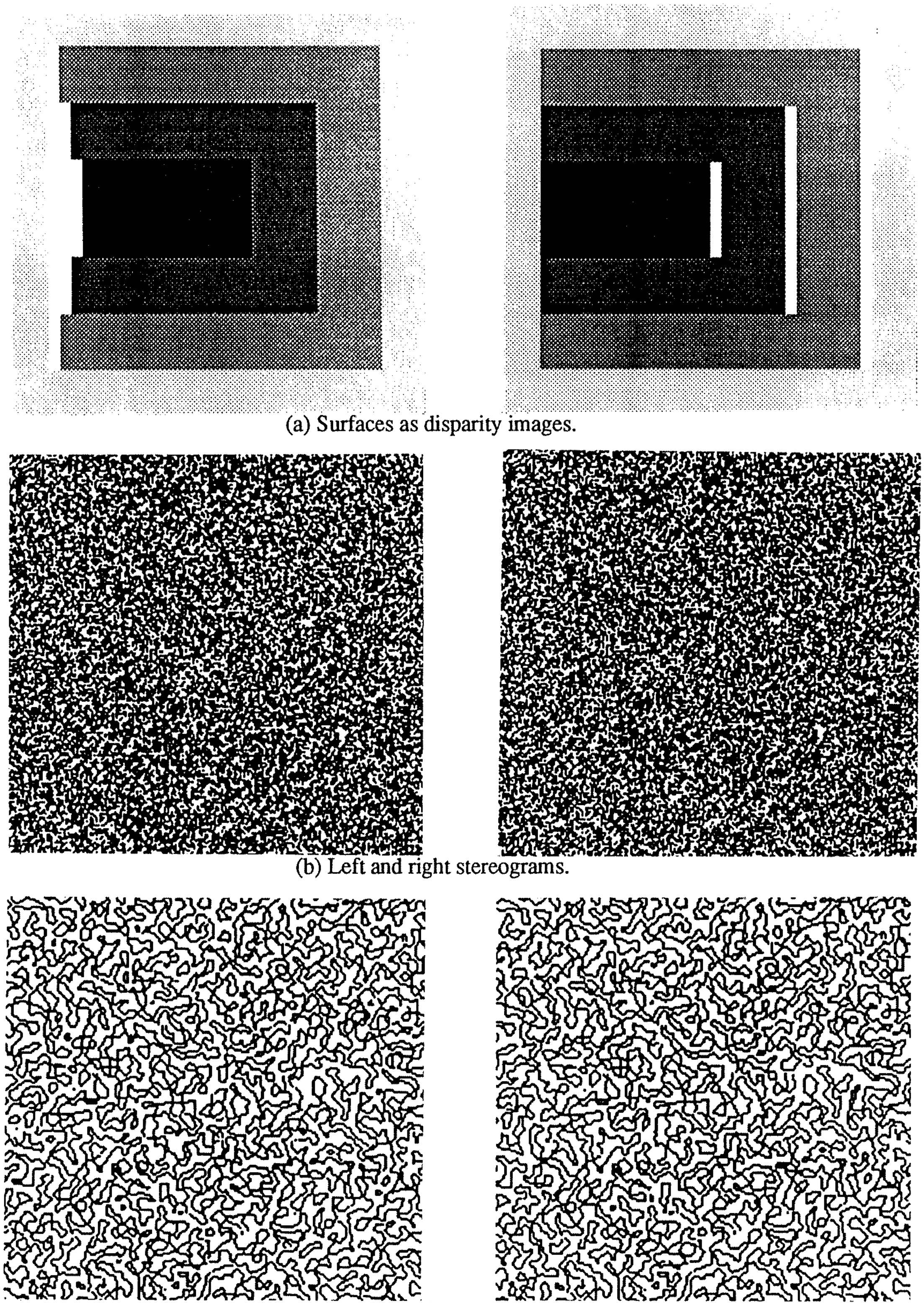
The General Support Algorithm uses integration of *local constraints* to produce high quality stereo matching. However, there are certain situations that the local constraints cannot explicitly address. For example, local constraints can not detect occluding boundaries and therefore continue to allow support across these boundaries. As we saw in the stereogram example, even though most occluded edges do not have matches, a large percentage of the errors made by the algorithm involved occlusions. Some of these problems can be solved through non-local matching mechanisms that feed back into the local network to improve the quality of matching.

The two most difficult problems for local matching constraints are:

- Structural differences between the two images. This involves occlusions, noise, and significant structural differences between the contours seen in the two images. Simple examples of each of these can generally be handled implicitly through the interactions of a number of the constraints. More significant differences cause a number of matching errors. In Figure 14, there are significant contour differences in the nearest parts of the sandwich. A more severe example of this is the books image (Figure 8) where some of the contours in the upper-right corner of the images appear almost arbitrarily.
- Repetitive textures that are either extremely dense or partially occluded. There is often little to distinguish between matches for these edges. One potential solution to these problems is to use an independent measure of depth such as the focal gradient. Such a coarse indicator of depth can favor the correct disparity enough that the ambiguity is resolved.

In addition to these general problems with local constraint interaction, our General Support Algorithm is sensitive to the density of edges in regions of the images. This problem is relatively minor except in highly ambiguous regions.

We are presently developing solutions to a number of these problems. The most important of these involves occlusion detection. This requires the use of additional nodes in the network. A group responds to average disparities in small region; other nodes can react to sharp changes between adjacent non-overlapping disparity nodes, thereby indicating an occlusion boundary. This can be used to eliminate disparity gradient support across occluding boundaries and in some cases directly suppress known occluded matches. A number of recent algorithms have integrated surface reconstruction with matching and have proposed using the result to locate occlusions.[8, 18] These approaches seem promising, but they do not incorporate significant local matching techniques. Thus, they require complete surface reconstruction to locate occlusions. However, this is not necessary, because disparity tends to vary slowly except at occlusions.



(a) Surfaces as disparity images.

(b) Left and right stereograms.

(c) Level 1 edges.

Figure 4. Stereogram example.

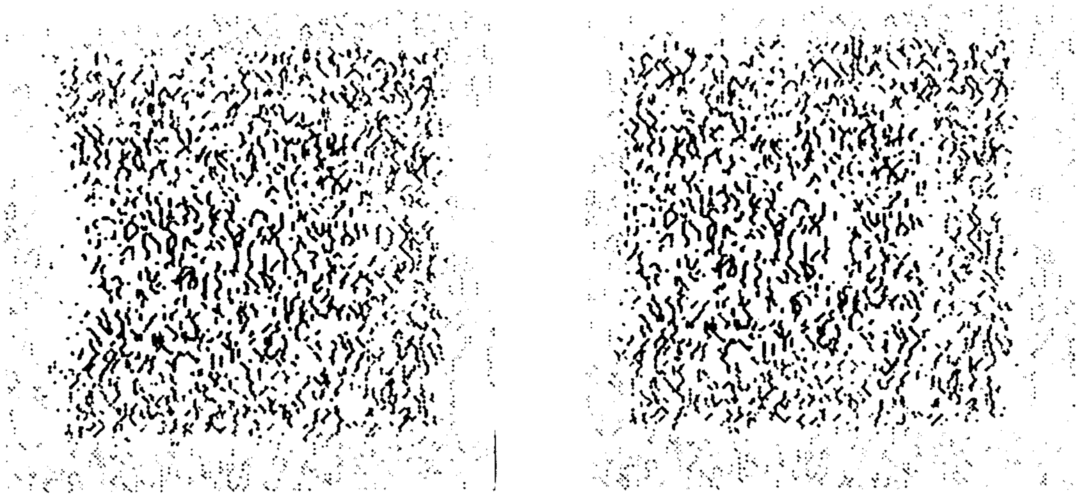


Figure 4 (d). Disparity images resulting from intermediate level matches.

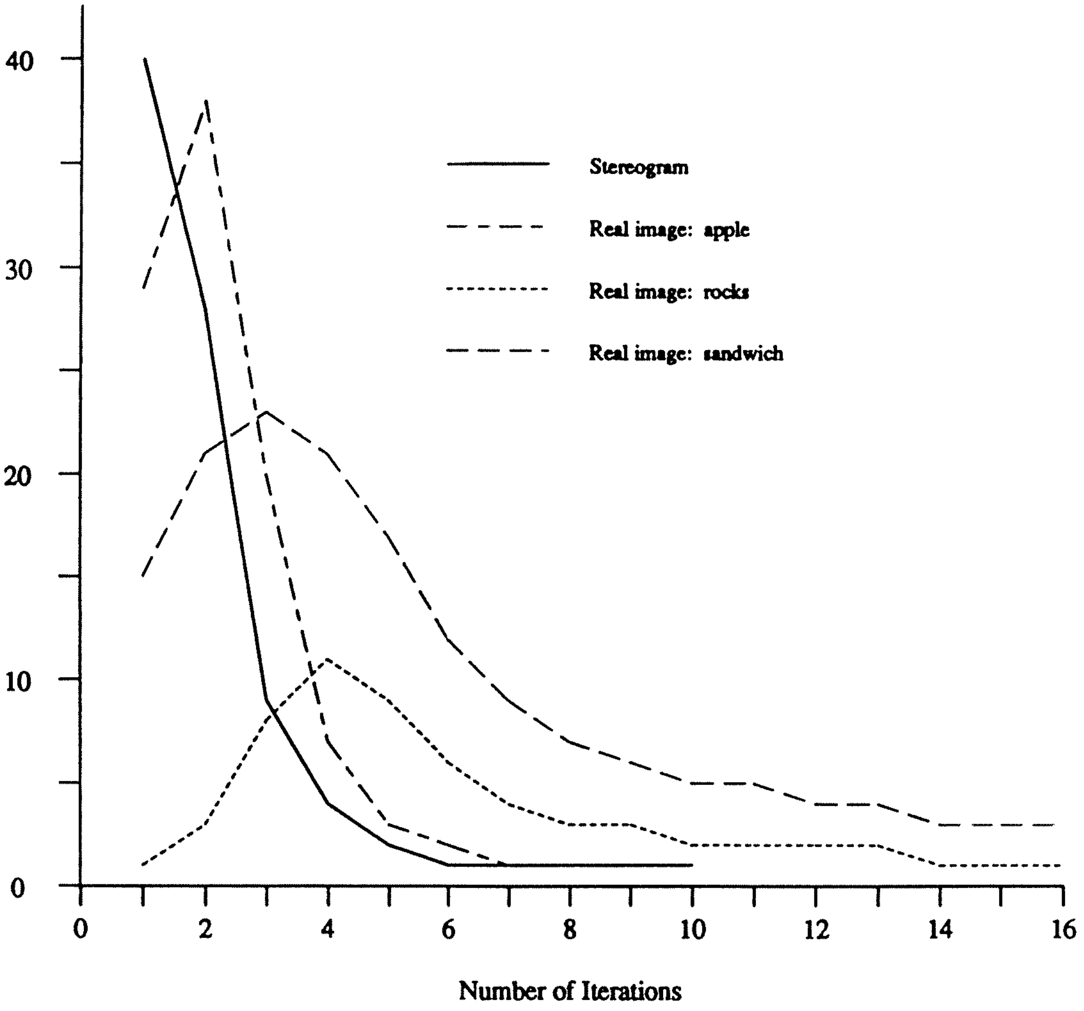
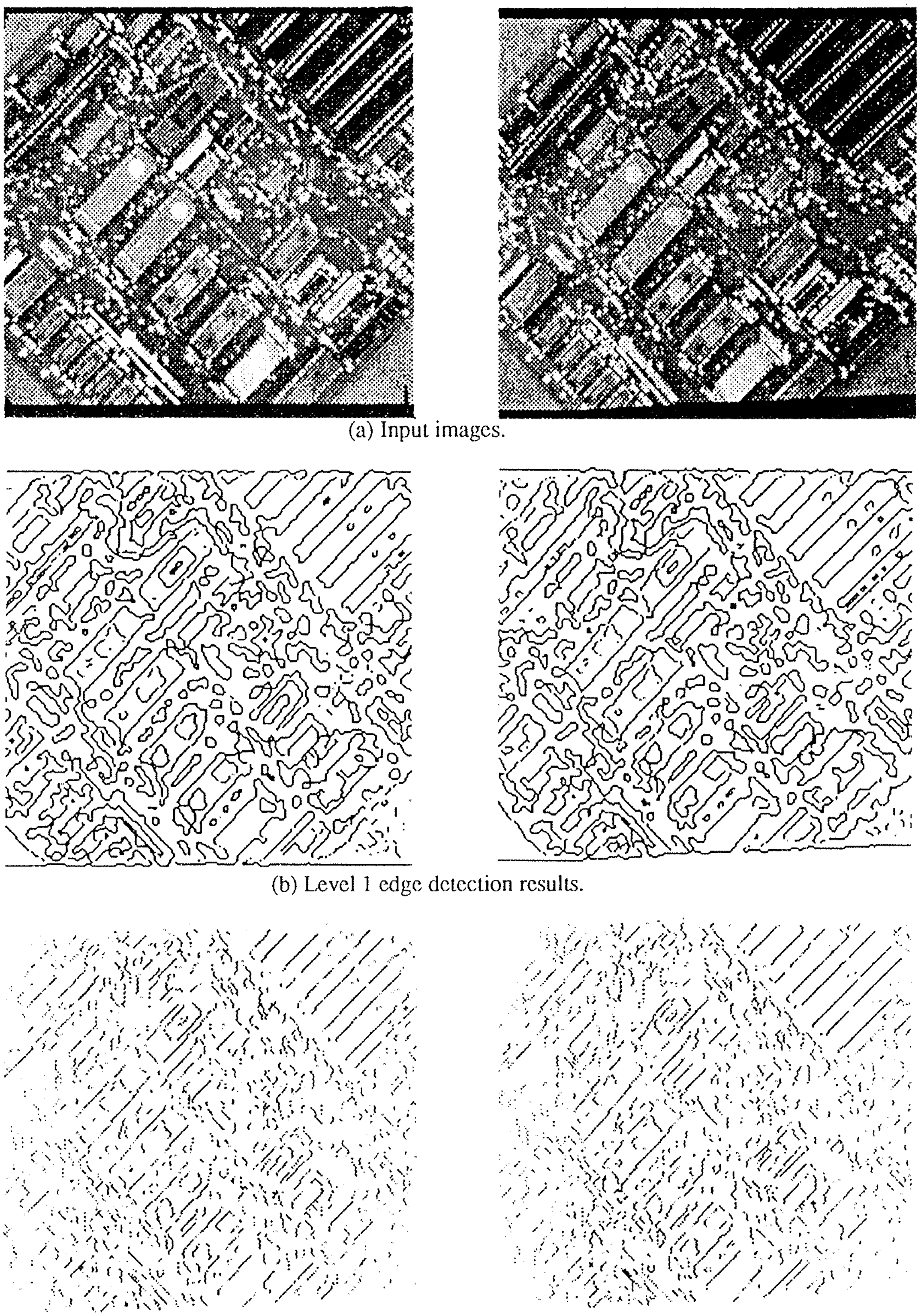


Figure 5. Percent of nodes in the intermediate level of activation as a function of the iteration number. Results are shown for the stereogram and for three real images.

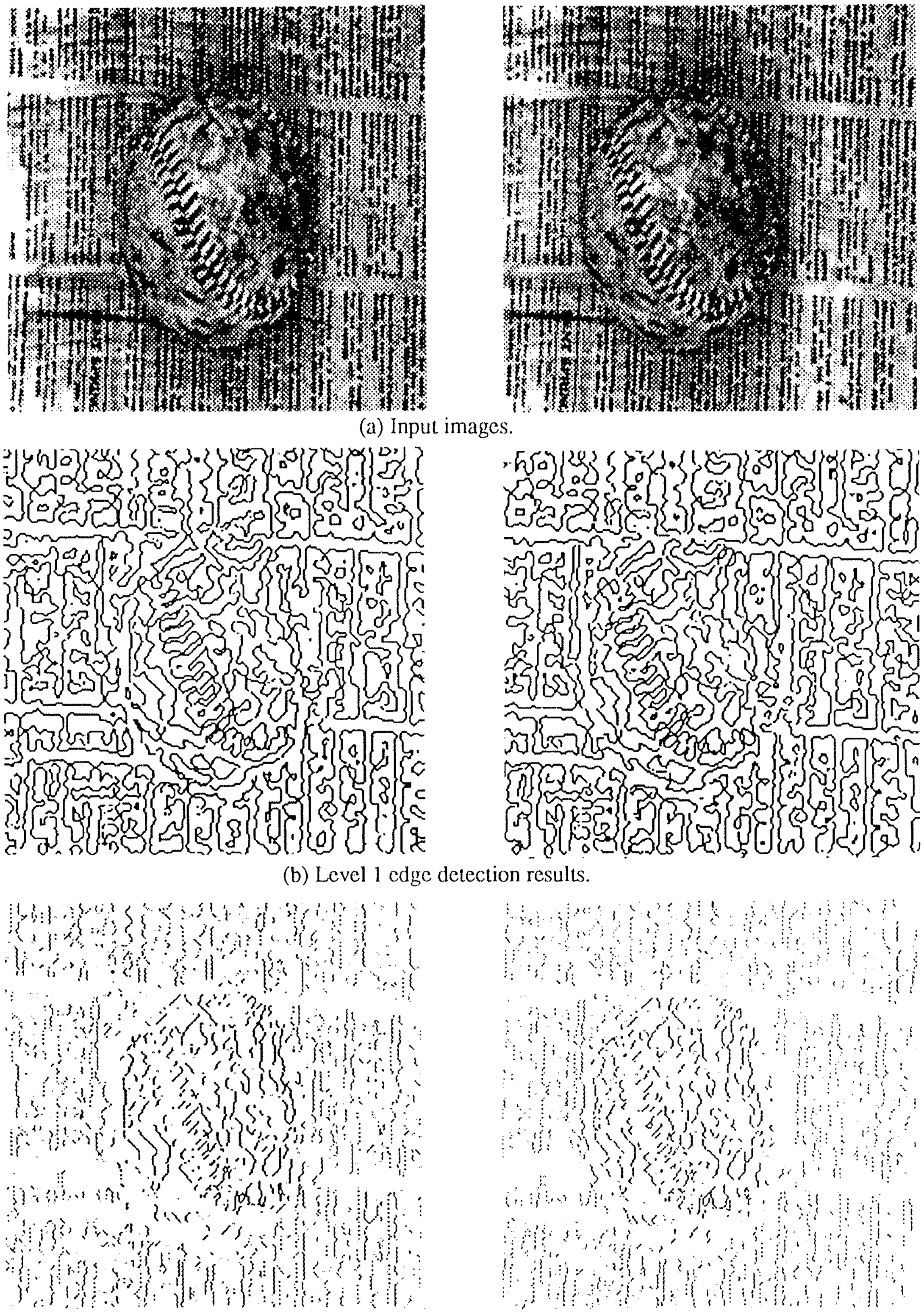


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 6. Apple processor board images. The disparity range between the images was small.

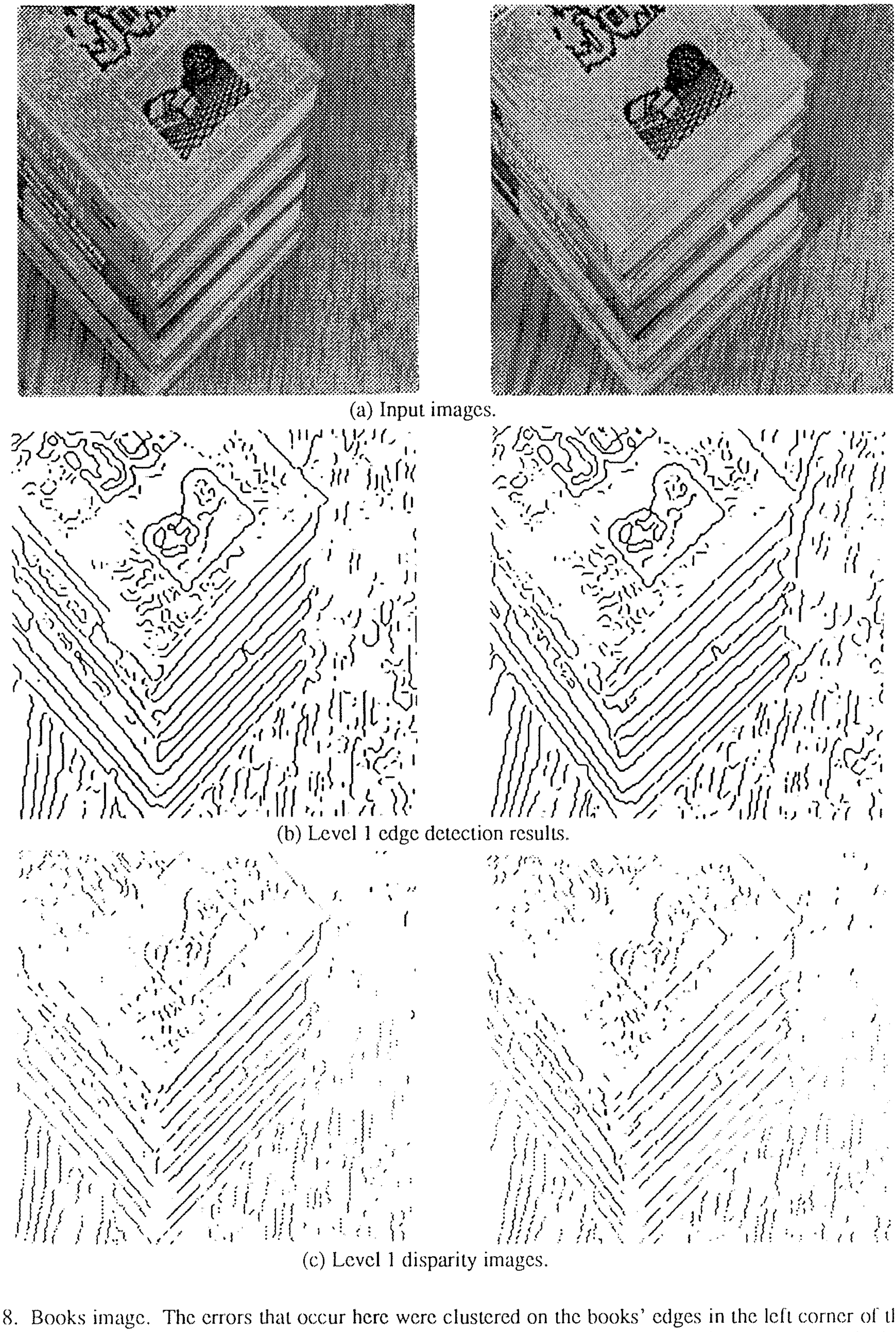


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 7. Image of a baseball in front of newsprint. The newsprint was all at approximately the same disparity. At this resolution the periodic pattern is easily resolved. The occluded areas for which no matches should be found were to the left and right of the baseball in the left and right images respectively.

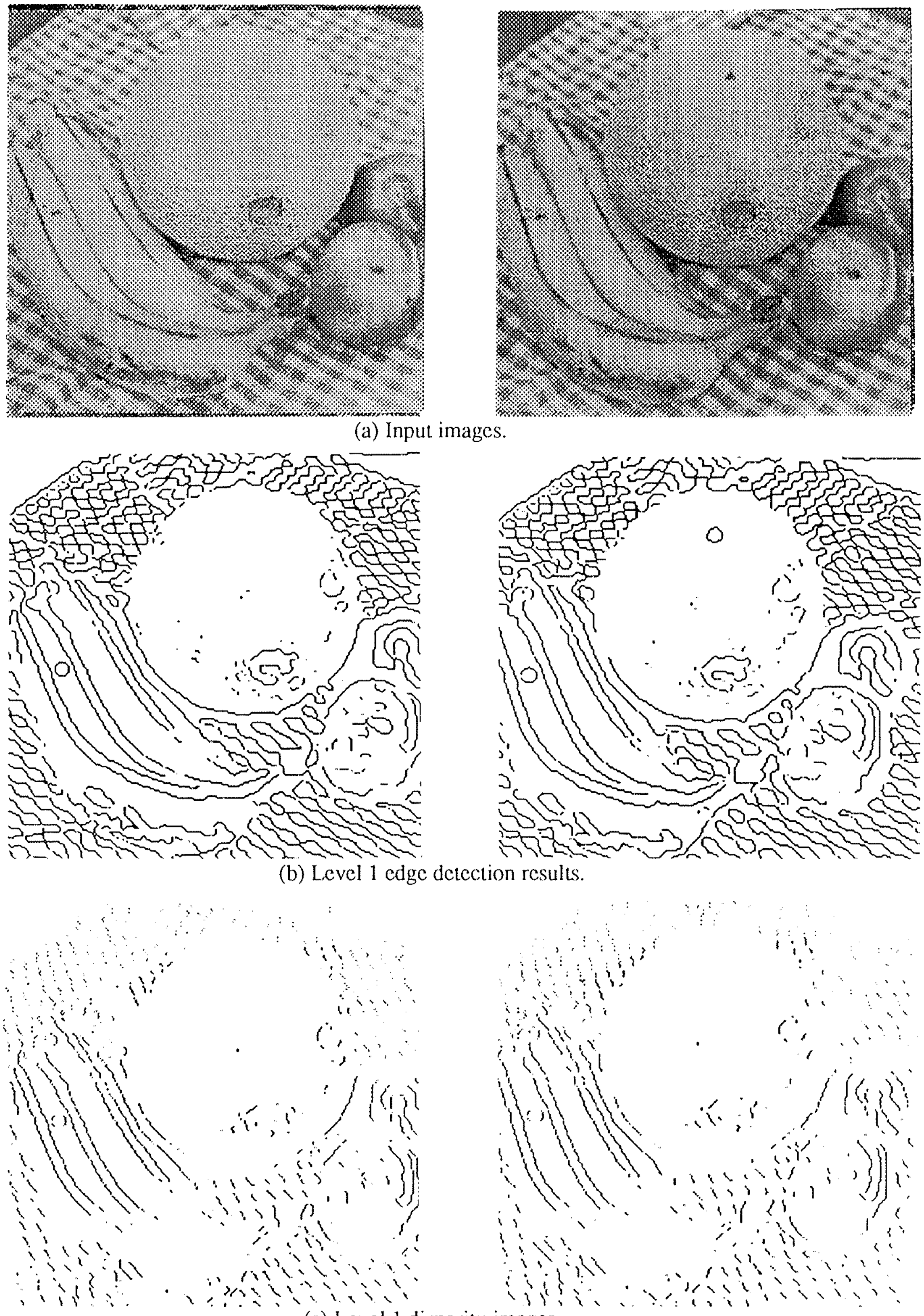


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 8. Books image. The errors that occur here were clustered on the books' edges in the left corner of the images, and on the table in the upper right part of the images. There were significant variations between the images in these places.

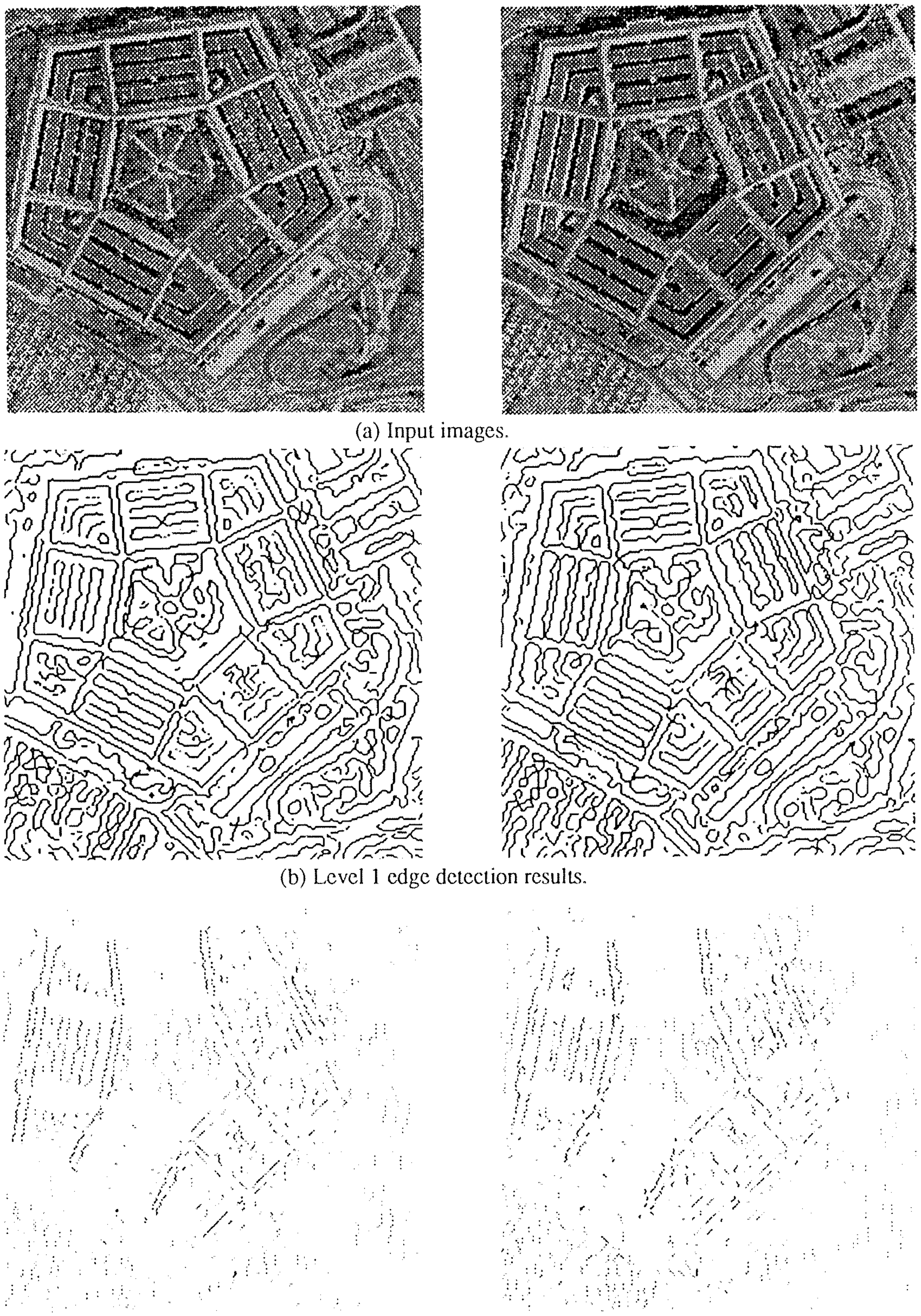


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 9. Fruit image. The matches on the fruit and on the tablecloth in front of the fruit were almost all correct. The errors occurred for edges from the tablecloth near the surface of the orange. The combination of the occluding region and the periodic structure of the table cloth produced incorrect matches for a number of edges.

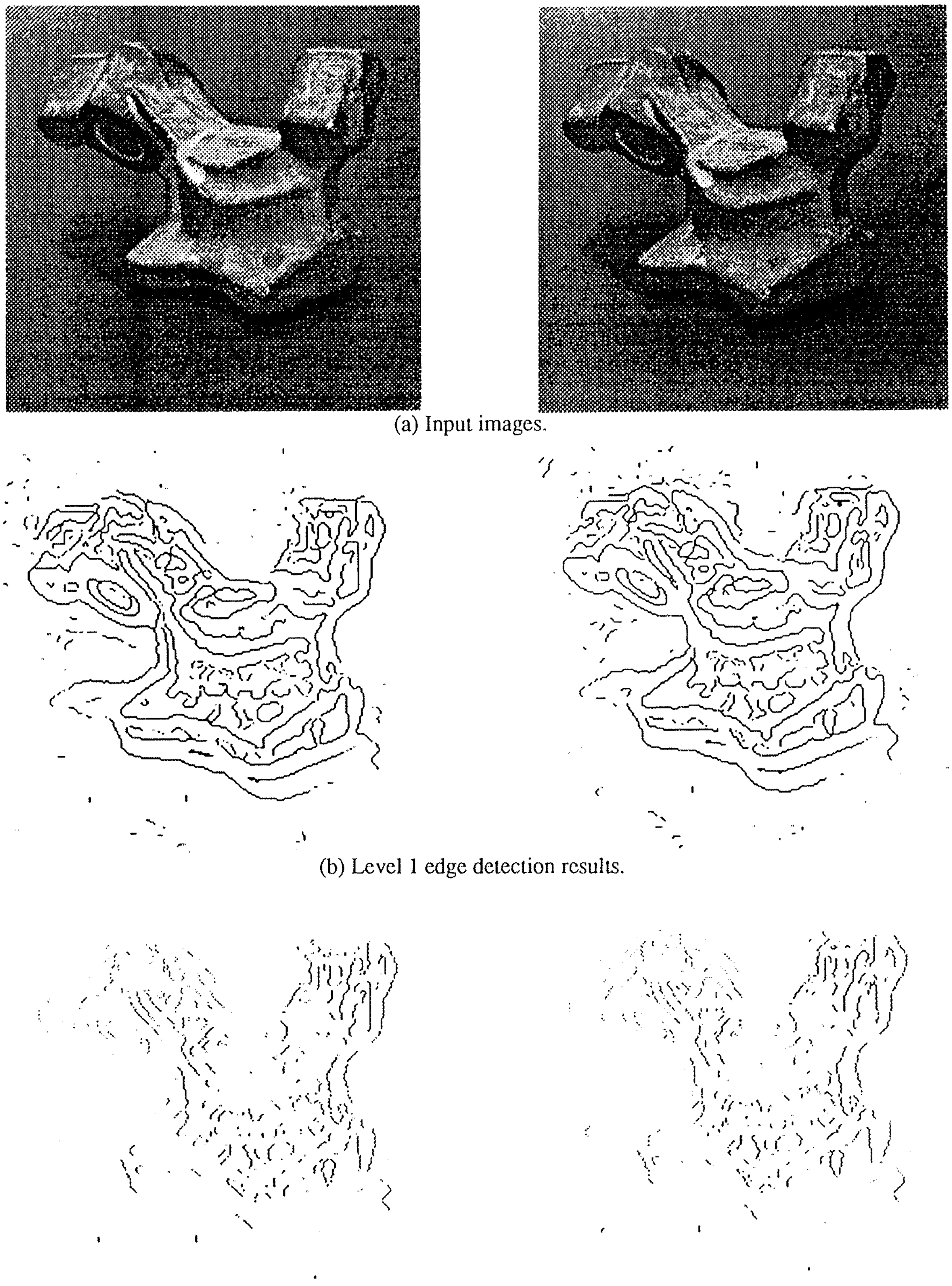


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 10. Pentagon image. There are some significant contour differences between the two images on the interior of the Pentagon. This causes matching confusion that the algorithm did not completely overcome.

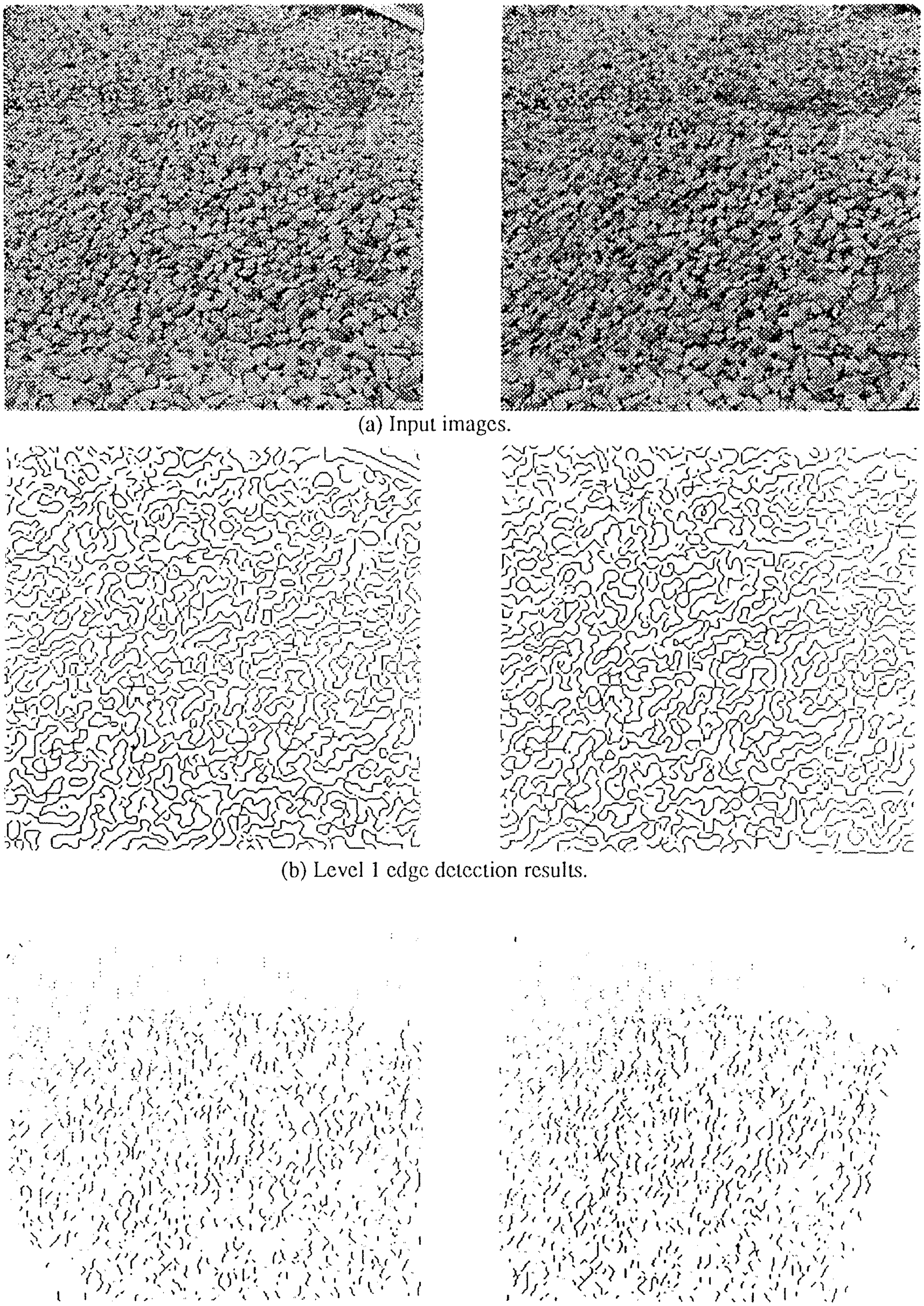


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 11. Renault autopart image. Almost all of the matches were correct. Errors only occurred sporadically when there were significant differences in the contours resulting from edge detection.

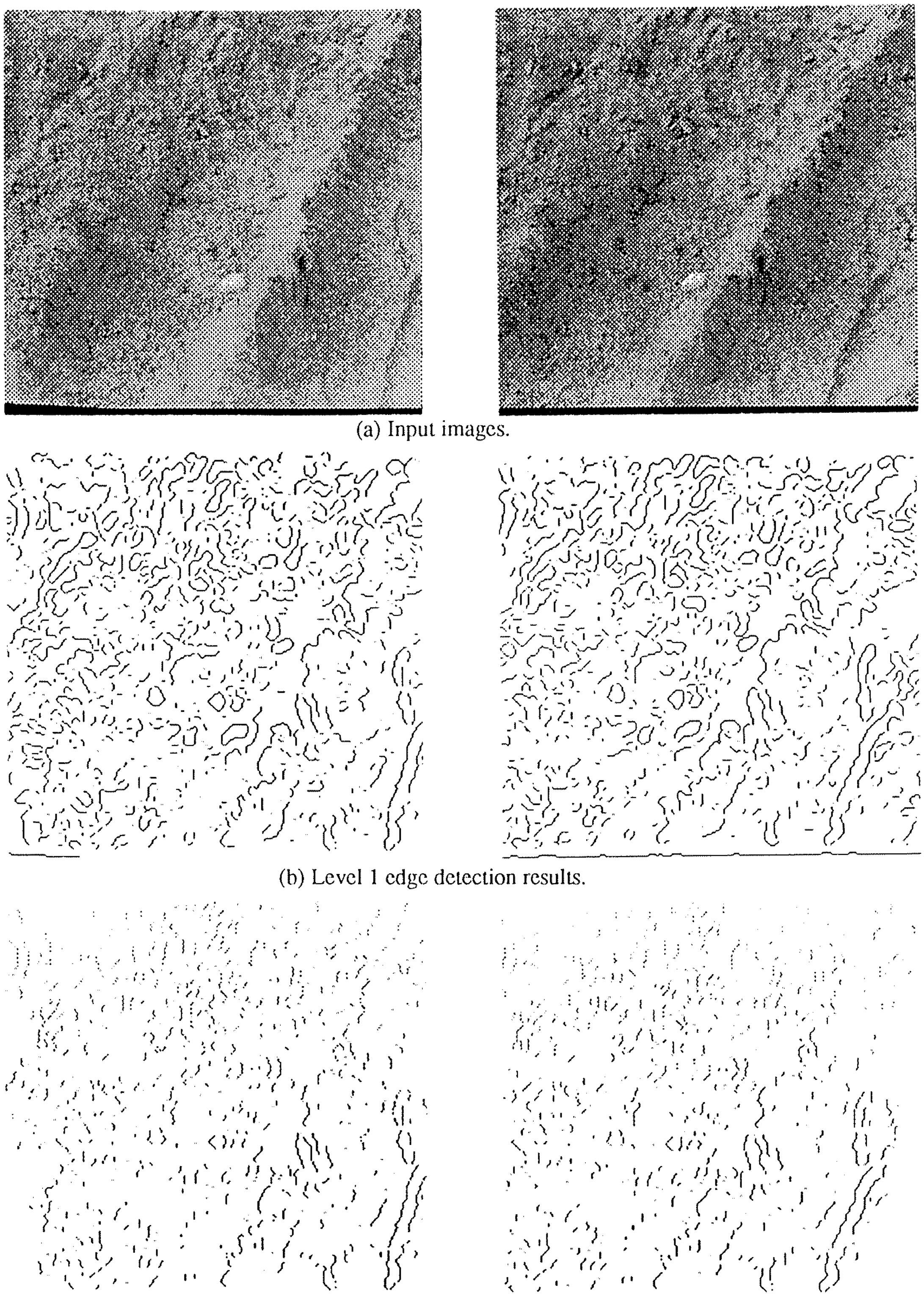


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 12. Rocks image. There was a large occlusion in the upper region of the images. The disparity range between the images was extremely large. Errors were sporadic.

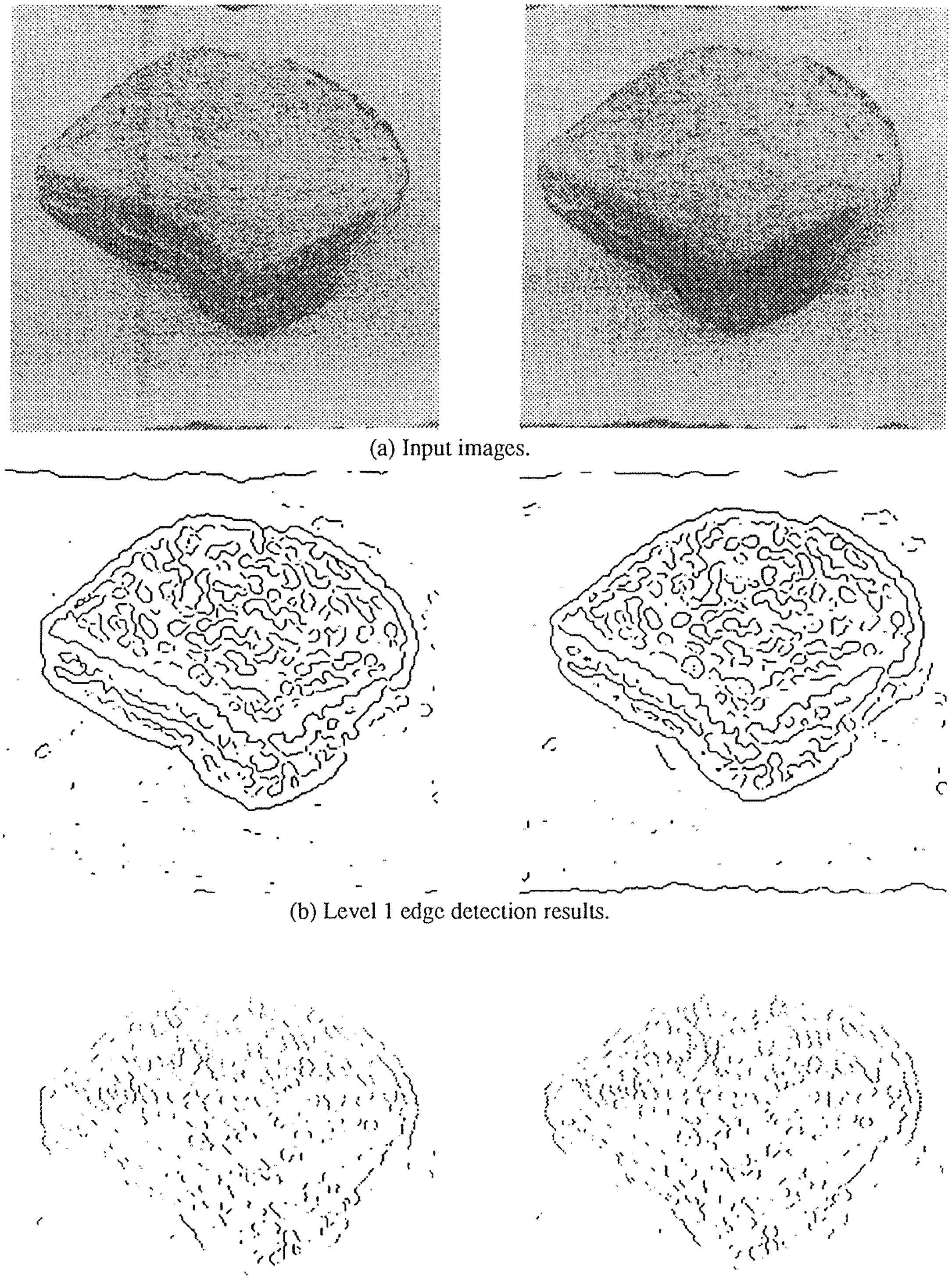


(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 13. Ruts image. There were almost no errors.



(a) Input images.

(b) Level 1 edge detection results.

(c) Level 1 disparity images.

Figure 14. Sandwich image. The appearance of the lower edges of the sandwich varies significantly between the two images. This caused a small number of errors.

References

1. H. Harlan Baker, T. O. Binford, J. Malik, and J-F. Meller, ''Progress in stereo matching,'' *Proc. DARPA Image Understanding Workshop*, pp. 327-335 (1983).

2. H. H. Baker and T. O. Binford, ''Depth from edge and intensity based stereo,'' *Proc. Seventh Int. Joint Conf. on Artificial Intelligence*, pp. 631-636 (1981).

3. S. T. Barnard and W. B. Thompson, ''Disparity analysis of images,'' *IEEE Trans. on Pattern Analysis and Machine Intelligence* **2** pp. 333-340 (1980).

4. S. T. Barnard and M. A. Fischler, ''Computational Stereo,'' *Computing Surveys* **14** pp. 553-572 (1982).

5. K.L. Boyer and A.C. Kak, ''Symbolic stereo from structural descriptions,'' *Proc. Conf. on Artificial Intelligence Applications*, pp. 82-87 (1985).

6. L. S. Davis and A. Rosenfeld, ''Cooperating processes for low-level vision: A survey,'' *Artificial Intelligence* **17** pp. 245-263 (1981).

7. M. Drumheller and T. Poggio, ''On parallel stereo,'' *Proc. IEEE Int. Conf. on Robotics and Automation* **3** pp. 1439-1448 (1986).

8. R. D. Eastman and A. M. Waxman, ''Using disparity functionals for stereo correspondence and surface reconstruction,'' *Computer Vision, Graphics, and Image Processing* **39** pp. 73-101 (1987).

9. J. A. Feldman and D. H. Ballard, ''Connectionist models and their properties,'' *Cognitive Science* **6** pp. 205-254 (1982).

10. A. Goshtasby, ''A refined technique for stereo depth perception,'' *Workshop on Computer Vision: Representation and Control*, pp. 125-129 (1984).

11. W.E.L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, Massachusetts (1981).

12. W.E.L. Grimson, ''Computational experiments with a feature based stereo algorithm,'' *IEEE Trans. on Pattern Analysis and Machine Intelligence* **7** pp. 17-34 (January, 1985).

13. S. Grossberg, ''Competitive learning: From interactive activation to adaptive resonance,'' *Cognitive Science* **11** pp. 23-64 (1987).

14. M. J. Hannah, ''SRI's baseline stereo system,'' *Proc. DARPA Image Understanding Workshop*, pp. 149-155 (December, 1985).

15. M. Herman, T. Kanade, and S. Kuroe, ''Incremental acquisition of a three-dimensional scene model from images,'' *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6** pp. 331-340 (1984).

16. E. C. Hildreth, ''Edge detection,'' MIT AI Laboratory A.I. Memo 858 (September, 1985).

17. W. Hoff and N. Ahuja, ''Surfaces from stereo,'' *Proc. DARPA Image Understanding Workshop*, pp. 98-106 (December, 1985).

18. W. Hoff and N. Ahuja, ''Extracting surfaces from stereo images: an integrated approach,'' *Proc. First Int. Conf. Computer Vision*, pp. 284-294 (1987).

19. J. J. Hopfield and D. W. Tank, ''Computing with neural circuits: A model,'' *Science* **233** pp. 625-633 (August 8, 1986).

20. R. A. Hummel and S. W. Zucker, ''On the foundations of relaxation labeling processes,'' *IEEE Trans. on Pattern Analysis and Machine Intelligence* 5(3) pp. 267-287 (1983).

21. B. Julesz, *Foundations of Cyclopean Perception*, University of Chicago Press, (1971).

22. Y. C. Kim and J.K. Aggarwal, ''Finding range from stereo images,'' *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 289-294 (1985).

23. H. S. Lim and T. O. Binford, ''Stereo correspondence: features and constraints,'' *Proc. DARPA Image Understanding Workshop*, pp. 373-380 (December, 1985).

24. D. Marr and T. Poggio, ''A computational theory of human stereo vision,'' *Proc. Royal Society of London, B.* **204** pp. 301-328 (1979).

25. D. Marr and E. Hildreth, ''Theory of edge detection,'' *Proc. Royal Society of London B.* **207** pp. 187-217 (1980).

26. D. Marr, *Vision,* W.H. Freeman and Company, New York (1982).

27. J. E.W. Mayhew and J. P. Frisby, "Psychophysical and computation studies towards a theory of human stereopsis," *Artificial Intelligence* **17** pp. 349-385 (1981).

28. G. Medioni and R. Nevatia, "Segment-based stereo matching," *Computer Vision, Graphics and Image Processing* **31** pp. 2-18 (1985).

29. Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **7** pp. 139-154 (1985).

30. Y. Ohta, K. Takano, and K. Ikeda, "A highspeed stereo matching system based on dynamic programming," *Proc. First Int. Conf. on Computer Vision,* pp. 335-342 (1987).

31. A. P. Pentland, "A new sense for depth of field," *Proc. Ninth Int. Joint Conf. on Artificial Intelligence,* pp. 988-994 (1985).

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

33. K. Prazdny, "Detection of binocular disparities," *Biological Cybernetics* **52** pp. 93-99 (1985).

34. G.V.S. Raju, T.O Binford, and S. Shekhar, "Stereo matching using viterbi algorithms," *Proc. DARPA Image Understanding Workshop,* pp. 766-776 (1987).

35. D. E. Rumelhart, G. E. Hinton, and J. L. McClelland, "A general framework for parallel distributed processing," pp. 45-76 in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations,* ed. D. E. Rumelhart and J. L. McClelland,The MIT Press, Cambridge, MA (1986).

36. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," pp. 318-362 in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations,* ed. D. E. Rumelhart and J. L. McClelland,The MIT Press, Cambridge, MA (1986).

37. D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," pp. 151-193 in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations,* ed. D. E. Rumelhart and J. L. McClelland,The MIT Press, Cambridge, MA (1986).

38. D. L. Smitley and R. Bajcsy, "Stereo processing of aerial, urban images," *Proc. Seventh IEEE Int. Conf. on Pattern Recognition* **1** pp. 433-435 (1984).

39. R. Szeliski and G. Hinton, "Solving random-dot stereograms using the heat equation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 284-288 (1985).

40. R. Szeliski, "Cooperative algorithms for solving random-dot stereograms," CMU-CS-86-133 (1986).

41. H. P. Trivedi, "A computational theory of stereo vision," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 277-282 (1985).

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

43. A. M. Waxman and J. H. Duncan, "Binocular image flows: steps toward stereo-motion fusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **8** pp. 715-729 (1986).

44. A. M. Waxman and S. S. Sinha, "Dynamic stereo: passive ranging to moving objects from relative image flows," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **8** pp. 406-412 (1986).

45. L. R. Williams, "Spectral continuity and eye vergence movements," *Proc. Ninth Int. Joint Conf. on Artificial Intelligence,* pp. 985-987 (1985).

46. A. P. Witkin, "Scale-space filtering," *Proc. Eighth Int. Joint Conf. on Artificial Intelligence,* pp. 1019-1022 (1983).

47. G. Xu, S. Tsuji, and M. Asada, "Coarse-to-fine control strategy for matching motion stereo pairs," *Proc. Ninth Int. Joint Conf. on Artificial Intelligence,* pp. 892-894 (1985).