# *New Algorithms for 3D Surface Description from Binocular Stereo using Integration*

*by* K. SUNIL KUMAR *and* U. B. DESAI

*Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology, Bombay* 400 076, *India*

ABSTRACT: *In this paper, we formulate and develop an approach which integrates different modules (feature extractor, matching and interpolation) involved in stereo. We study the integration process at the finest resolution when (i) the precomputed edge map is the only line field driving the model, (ii) the line fields are computed interactively by the feature extracting module of the model, and (iii) when both the interactive line field computation module and the precomputed line field modules are present. This integration process being computationally intensive, we develop a multiresolution stereo integration approach. The energy function for each module at different resolutions is constructed and minimized in an integrated manner yielding a dense disparity map. A new energy function for the matching module is proposed. Experimental results are presented to illustrate our approach.*

## *I. Introduction*

Vision comes to humans so naturally that one does not realize its complexity until one tries to automate it. The basic aim of stereo vision is to extract the disparity map or equivalently the depth map (provided the camera geometry is known) of the scene from its two-dimensional representation (**1**). Obtaining depth information from the 2D stereo images is an inverse optics problem and hence ill-posed in the sense of Hadamard (**2**). Regularization in the sense of Tikhonov is a commonly used technique in computer vision problems to make an ill-posed problem well-posed. The basic philosophy of this technique is to restrict the solution space by imposing some constraints. Uniqueness, smoothness, epipolar geometry and disparity gradient are some of the physical constraints that are imposed on the stereo problem to restrict the solution space and hence make the problem well-posed.

Numerous techniques have been developed to infer depth [(**3**–**11**) to cite a few]. Various shape from X techniques get the depth information by using only a single 2D image, but they make use of a number of assumptions like distant light sources, spherical patches, knowledge of the reflectance map, etc. In addition, the depth information obtained is relative and not absolute. Binocular stereo techniques use two 2 D images (of the same scene) to extract depth which is absolute.

Though there exists a large number of stereo vision algorithms, most of them



0016-0032(95)00008-9

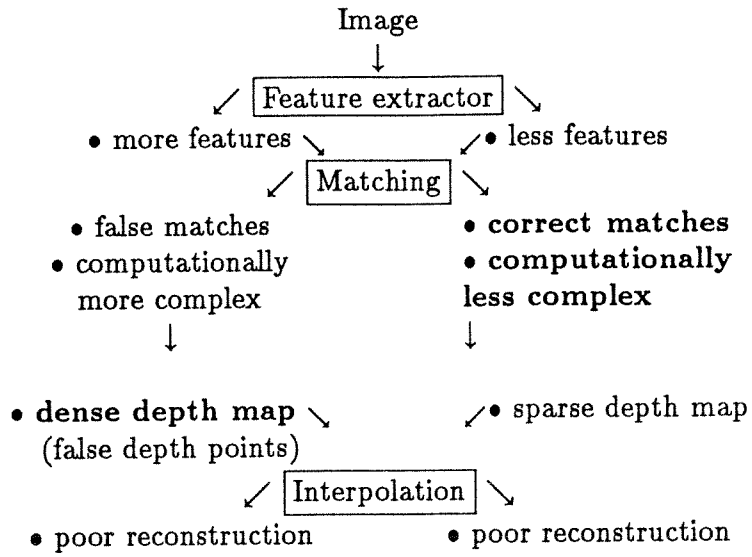


FIG. 1. Procedure involved in a typical stereo vision algorithm.

can be categorized as having three main modules built into their structure. They are

(1) Feature extraction module,
(2) Matching module,
(3) Interpolation module.

A number of physical constraints are imposed in the development of most algorithms. Also, in most of the algorithms the three blocks are traversed in a top-to-bottom fashion without any interaction between the modules. Initially a feature extracting algorithm is used to obtain salient features in both the left and the right image of the stereo pair. In the second stage, a matching algorithm is used to match the feature points in the left and the right image. Once the feature points are matched, the difference between the position of the feature point in the left image and the corresponding matched point in the right image gives the disparity map. Knowing the geometry of the camera, the depth information can be calculated from the disparity map in a simple way. Next an interpolation algorithm is used to interpolate on the sparse depth data, to obtain a dense depth map (in our simulations we interpolate on the disparity data rather than the depth data since the camera geometry is usually not known).

The problem emanating from large and small numbers of feature points is depicted pictorially in Fig. 1. One would like to have a large number of correct matches, leading to a dense disparity map and consequently to a good reconstruction. However, the dense depth map and correct matches are conflicting (notice that they appear on different sides of the matching block in Fig. 1); one alternative is to use an integration approach. Integration is a process where information obtained from one module drives another module with the aim of improving the final solution. A number of algorithms appear in the vision literature which use integration in one form or another [for example (12–18)].

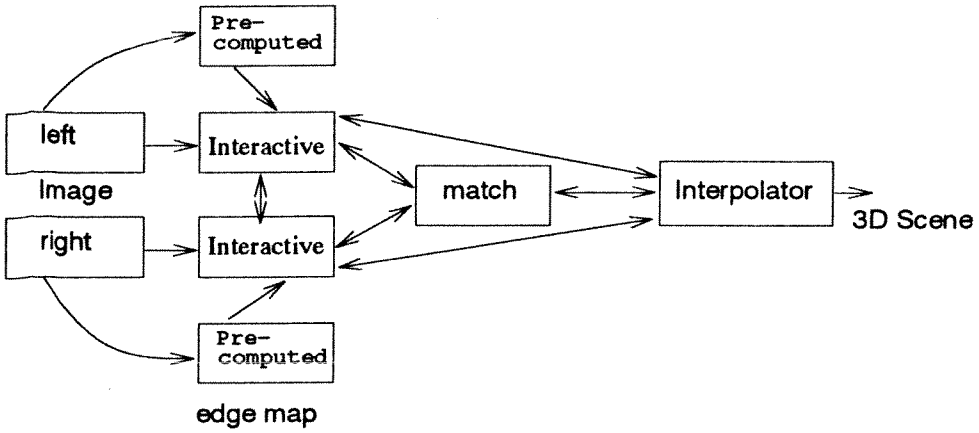


FIG. 2. The integration model.

Perhaps one could view the early work of Moravec (**19**) as an approach to integrating information from several images of the same scene from different views. More recently the Markov random field (MRF) framework has been exploited to integrate information from multiple cues like motion, color and texture (**12**). Another approach integrates the stereo pair of images at different scales, the information available at the coarser pairs is used to aid the images at finer scales (**16**). Integration of stereo images at various baselines has been worked out in (**15**).

In this paper, motivated by the work of Gamble *et al.* (**12**), Clark and Yullie (**18**), Toborg and Hwang (**13**) and Barnard (**16**) we present a novel approach to integrate various modules associated with stereo. In our approach there is integration among the following blocks: right and left edge extracting modules, the matching module, and the interpolator module (see Fig. 2). We also incorporate line fields to preserve discontinuities in the reconstructed surface. Moreover, the interaction between modules is through these line fields. Our integration approach results in a disparity map which is dense. The computation of the dense disparity map from two 2D stereo images is computationally expensive. To overcome the problem of high computational time we have exploited the multiresolution approach of Burt and Adelson (**20**) in our integration model. We would like to remark that the multiresolution approach of (**20**) is used very extensively in the computer vision literature [for example (**14**, **16**, **21**–**23**)]. As shown in Section VII the multiresolution approach to integrated stereo is at least seven and a half times faster than the one without multiresolution. Besides faster computation, our simulation studies illustrate that a more accurate disparity map is obtained by exploiting multiresolution (see Section VII). As explained in Sections VI and VII, this is because, in the multiresolution approach, we have a better disparity initialization for the disparity computation at the highest resolution as compared to zero initialization when one works on the single finest resolution.

## *II. The Integration Model*

Our stereo integration model is depicted in Fig. 2. As seen from the figure there is feedforward as well as feedback (strong) interaction among the following

Vol. 331B, No. 5, pp. 531–554, 1994

modules: the interactive left and right edge detectors, the matching module, and the interpolation module. There is only feedforward (weak) interaction between the precomputed edge map and the interactive edge detectors. The interaction model structure is analogous to Clark and Yullie's (**18**) recurrent interactive model, except that we also have strong interaction with the interpolation module. Interaction between modules is achieved through line fields, which also facilitate preserving discontinuities in the reconstructed surface. In the multiresolution framework, Fig. 2 represents the interaction model at any given resolution $\Omega$. The specifics of the interaction achieved between different modules is explicated in Section IV. Though we use line fields (edges) for interaction in this paper, it is conceivable that some other feature amenable to interaction may be used.

## *III. Problem Formulation*

Let $z_{m,n}$ be the scene under consideration. Let $z_{i,j}$ represent the depth value of the scene at the point $(i,j)$, and $(i,j)$ represent a point on the 2D lattice of size $P \times P$ (where $P = 2^{\Omega}$; $\Omega$ is an integer, $0 \leqslant m, n \leqslant (P-1)$). Let $x_{m,n}^{R,\Omega}, x_{m,n}^{L,\Omega}$ be the 2D intensity images of the scene $z_{m,n}$, obtained by the left and the right disparate cameras respectively. The problem of stereo vision is to estimate $z_{m,n}$ given the camera geometry, $x_{m,n}^{L,\Omega}$ and $x_{m,n}^{R,\Omega}$ for $0 \leqslant m, n \leqslant (2^{\Omega}-1)$. Since the camera geometry is not available, in our formulation we consider the estimation of the disparity $\{d_{m,n}\}$.

### 3.1. *Notation used*

At resolution $\Omega$,

- $x_{i,j}^{R,\Omega}$ is the intensity value of the $(i,j)$th pixel in the right image,
- $x_{i,j}^{L,\Omega}$ is the intensity value of the $(i,j)$th pixel in the left image,
- $v_{i,j}^{L,\Omega}$ is the vertical line field between pixels $(i,j)$ and $(i,j-1)$ for the interactive left image edge detector,
- $v_{i,j}^{R,\Omega}$ is the vertical line field between pixels $(i,j)$ and $(i,j-1)$ for the interactive right image edge detector,
- $h_{i,j}^{L,\Omega}$ is the horizontal line field between pixels $(i,j)$ and $(i-1,j)$ for the interactive left image edge detector,
- $h_{i,j}^{R,\Omega}$ is the horizontal line field between pixels $(i,j)$ and $(i-1,j)$ for the interactive right image edge detector,
- $\bar{h}_{i,j}^{L,\Omega}$ is the horizontal line field between pixels $(i,j)$ and $(i-1,j)$ for the precomputed left image edge detector,
- $\bar{h}_{i,j}^{R,\Omega}$ is the horizontal line field between pixels $(i,j)$ and $(i-1,j)$ for the precomputed right image edge detector,
- $\bar{v}_{i,j}^{L,\Omega}$ is the vertical line field between pixels $(i,j)$ and $(i,j-1)$ for the precomputed left image edge detector,
- $\bar{v}_{i,j}^{R,\Omega}$ is the vertical line field between pixels $(i,j)$ and $(i,j-1)$ for the precomputed right image edge detector,
- $d_{i,j}^{\Omega}$ is the disparity at $(i,j)$th pixel,
- $h_{i,j}^{I,\Omega}$ is the horizontal line field in the interpolated scene.

- $v_{i,j}^{d,\Omega}$ is the vertical line field in the interpolated scene,
- all line fields take values 1 or 0,
- $\alpha, \beta, \lambda, \gamma$ with all subscripts are real constants, and $\oplus$ is the binary operator such that $a \oplus b = 1$ only if $a = b = 0$, where $a$, $b$ take binary values 0 or 1, and
- we assume horizontal epipolar line constraint so that there is disparity only in the $j$ direction.

We proceed by constructing an energy function for each module which, while achieving the requirement of the module when minimized, also integrates information available from other modules, so as not to overlook the outcome of the other modules. This integration is achieved through the use of line fields. Though not explicitly stated, there is an obvious Markov random field (MRF) model underlying each energy function. For further details on line fields see Geman and Geman (**24**).

The energy functions are constructed and a brief description of the terms appearing in each energy function is given below. While constructing the energy function we assume that the stereo pair is neither noisy nor blurred (one can, without much difficulty, add an extra term to the energy function to take care of noise in the stereo image pair). The energy functions appearing in Section IV are constructed at resolution $\Omega$ and the energy functions at other resolutions can, for example at resolution $\Omega-1$, be obtained by replacing $\Omega$ with $\Omega-1$.

## *IV. The Energy Functions*

### 4.1. *The interactive left image edge detector*

We write

$$
\begin{aligned}
U_{edge}^{L,\Omega}(v^{L,\Omega}, h^{L,\Omega}) = \sum_i \sum_j [ & \lambda_1^{\Omega}\{(x_{i,j}^{L,\Omega} - x_{i,j-1}^{L,\Omega})^2(1-v_{i,j}^{L,\Omega}) + (x_{i,j}^{L,\Omega} - x_{i-1,j}^{L,\Omega})^2(1-h_{i,j}^{L,\Omega})\} \\
& + \lambda_2^{\Omega}\{v_{i,j}^{L,\Omega} + h_{i,j}^{L,\Omega}\} \\
& + \lambda_3^{\Omega}\{(1-h_{i,j}^{L,\Omega})h_{i,j+d_{i,j}^{\Omega}}^{R,\Omega} + (1-h_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})h_{i,j}^{L,\Omega} + (1-v_{i,j}^{L,\Omega})v_{i,j+d_{i,j}^{\Omega}}^{R,\Omega} + (1-v_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})v_{i,j}^{L,\Omega}\} \\
& + \lambda_4^{\Omega}\{(1-h_{i,j}^{L,\Omega})h_{i,j}^{d,\Omega} + (1-h_{i,j}^{d,\Omega})h_{i,j}^{L,\Omega} + (1-v_{i,j}^{L,\Omega})v_{i,j}^{d,\Omega} + (1-v_{i,j}^{d,\Omega})v_{i,j}^{L,\Omega}\} \\
& + \lambda_5^{\Omega}\{(1-h_{i,j}^{L,\Omega})\bar{h}_{i,j}^{L,\Omega} + (1-\bar{h}_{i,j}^{L,\Omega})h_{i,j}^{L,\Omega} + (1-v_{i,j}^{L,\Omega})\bar{v}_{i,j}^{L,\Omega} + (1-\bar{v}_{i,j}^{L,\Omega})v_{i,j}^{L,\Omega}\} \\
& + \lambda_6^{\Omega}\{(1-\bar{h}_{i,j}^{L,\Omega})h_{i,j+d_{i,j}^{\Omega}}^{R,\Omega} + (1-h_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})h_{i,j}^{L,\Omega} + (1-\bar{v}_{i,j}^{L,\Omega})v_{i,j+d_{i,j}^{\Omega}}^{R,\Omega} + (1-v_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})\bar{v}_{i,j}^{L,\Omega}\} \\
& + \lambda_7^{\Omega}\{(1-h_{i,j}^{d,\Omega})\bar{h}_{i,j}^{L,\Omega} + (1-\bar{h}_{i,j}^{L,\Omega})h_{i,j}^{d,\Omega} + (1-v_{i,j}^{d,\Omega})\bar{v}_{i,j}^{L,\Omega} + (1-\bar{v}_{i,j}^{L,\Omega})v_{i,j}^{d,\Omega}\}]. \qquad (1)
\end{aligned}
$$

The first term in (1) is the usual *discontinuous regularization* term because of the line field terms $(1-v_{i,j}^{L,\Omega})$ and $(1-h_{i,j}^{L,\Omega})$. This term is to take care of discontinuities. If there is a discontinuity, namely the gradient $(x_{i,j}^{L,\Omega} - x_{i,j-1}^{L,\Omega})^2$ becomes large, then $v_{i,j}^{L,\Omega}$, the line function, takes a value of 1 and hence there is a reduction in the cost contributed by the first term of (1). But then the cost contributed by this term would be a minimum assuming discontinuities everywhere. To overcome this we

 

introduce a penalty term, namely the second term of (1), which adds to the cost whenever a discontinuity is detected. The first two terms suggest that if the discontinuity is significant then it is cheaper to introduce a line rather than trying to interpolate between the two pixels (meaning, if $\lambda_1^\Omega (x_{i,j}^{L,\Omega} - x_{i,j-1}^{L,\Omega})^2 > \lambda_2^\Omega$ then introducing a discontinuity costs less than trying to interpolate). The choice of $\lambda_1^\Omega$ and $\lambda_2^\Omega$ depend on what variation in pixel intensity values would be significant enough to assume the presence of an edge.

The third, fourth, fifth, sixth and seventh terms are the integration terms. The idea behind constructing these terms is the *belief* that if there is a feature in the right image then there should exist a corresponding feature in the left image, and the fact that the function $f(x,y) \triangleq x(1-y)+y(1-x)$ takes a minimum value (in this case 0) only when $x$ and $y$ agree with each other, namely $x = 0 = y$ or $x = 1 = y$; in our case the feature is a line field or equivalently an edge.

*Remark*

On occasions, we shall use the terms *line field* and *edge* interchangeably. Thus, in this paper, phrases like *precomputed edge map* and *precomputed line field* mean the same thing.

The third, fourth, fifth, sixth and seventh terms adhere to the above belief. As an example consider the third term in (1). First, this term assumes the availability of the disparity map $d_{i,j}^\Omega$. Now if there is a feature $h_{i,j}^{L,\Omega}$ in the left image, then we expect a corresponding feature in the right image at pixel location $(i,j+d_{i,j}^\Omega)$, namely the feature $h_{i,j+d_{i,j}^\Omega}^{R,\Omega}$. Similar arguments hold for the vertical line fields. Thus the third term will be minimum when indeed there is a proper correspondence between $h_{i,j}^\Omega, h_{i,j+d_{i,j}^\Omega}^{R,\Omega}$, and between $v_{i,j}^\Omega, v_{i,j+d_{i,j}^\Omega}^{R,\Omega}$. The terms $\lambda_3^\Omega, \lambda_4^\Omega, \lambda_5^\Omega, \lambda_6^\Omega$ and $\lambda_7^\Omega$ can be looked upon as parameters that determine the degree of influence of the other modules on the present module.

The third term integrates information obtained from the right edge detector $(v^{R,\Omega}, h^{R,\Omega})$ and the matching block $(d^\Omega)$; this integration is achieved via $h_{i,j+d_{i,j}^\Omega}^{R,\Omega}$ and $v_{i,j+d_{i,j}^\Omega}^{R,\Omega}$. Analogously, the fourth term integrates information obtained from the interpolation block $(v^{d,\Omega}, h^{d,\Omega})$ and the matching block $(d^\Omega)$. The effect of the third and fourth terms is to re-enforce edges in the left image depending on the presence or absence of edges in the right image and the interpolated scene. The fifth, sixth and seventh terms are driven by the presence or absence of precomputed edges in the left image.

4.2. *The interactive right image edge detector*

We write

$$U_{edge}^{R,\Omega}(v^{R,\Omega}, h^{R,\Omega}) = \sum_i \sum_j [\beta_1^\Omega \{(x_{i,j}^{R,\Omega} - x_{i,j-1}^{R,\Omega})^2 (1 - v_{i,j}^{R,\Omega}) + (x_{i,j}^{R,\Omega} - x_{i-1,j}^{R,\Omega})^2 (1 - h_{i,j}^{R,\Omega})\}$$

$$+ \beta_2^\Omega \{v_{i,j}^{R,\Omega} + h_{i,j}^{R,\Omega}\}$$

$$+ \beta_3^\Omega \{(1 - h_{i,j}^{R,\Omega}) h_{i,j-d_{i,j}^\Omega}^{L,\Omega} + (1 - h_{i,j-d_{i,j}^\Omega}^{L,\Omega}) h_{i,j}^{R,\Omega} + (1 - v_{i,j}^{R,\Omega}) v_{i,j-d_{i,j}^\Omega}^{L,\Omega} + (1 - v_{i,j-d_{i,j}^\Omega}^{L,\Omega}) v_{i,j}^{R,\Omega}\}$$

$$+ \beta_4^\Omega \{(1 - h_{i,j}^{R,\Omega}) h_{i,j-d_{i,j}^\Omega}^{d,\Omega} + (1 - h_{i,j-d_{i,j}^\Omega}^{d,\Omega}) h_{i,j}^{R,\Omega} + (1 - v_{i,j}^{R,\Omega}) v_{i,j-d_{i,j}^\Omega}^{d,\Omega} + (1 - v_{i,j+d_{i,j}^\Omega}^{d,\Omega}) v_{i,j}^{R,\Omega}\}$$

$$+\beta_5^{\Omega}\{(1-\bar{h}_{i,j}^{R,\Omega})h_{i,j}^{R,\Omega}+\bar{h}_{i,j}^{R,\Omega}(1-h_{i,j}^{R,\Omega})+(1-\bar{v}_{i,j}^{R,\Omega})v_{i,j}^{R,\Omega}+\bar{v}_{i,j}^{R,\Omega}(1-v_{i,j}^{R,\Omega})\}$$
$$+\beta_6^{\Omega}\{(1-\bar{h}_{i,j}^{R,\Omega})h_{i,j-d_{i,j}^{\Omega}}^{L,\Omega}+(1-h_{i,j-d_{i,j}^{\Omega}}^{L,\Omega})\bar{h}_{i,j}^{R,\Omega}+(1-\bar{v}_{i,j}^{R,\Omega})v_{i,j-d_{i,j}^{\Omega}}^{L,\Omega}+(1-v_{i,j-d_{i,j}^{\Omega}}^{L,\Omega})\bar{v}_{i,j}^{R,\Omega}\}$$
$$+\beta_7^{\Omega}\{(1-\bar{h}_{i,j}^{R,\Omega})h_{i,j-d_{i,j}^{\Omega}}^{d,\Omega}+(1-h_{i,j-d_{i,j}^{\Omega}}^{d,\Omega})\bar{h}_{i,j}^{R,\Omega}+(1-\bar{v}_{i,j}^{R,\Omega})v_{i,j-d_{i,j}^{\Omega}}^{d,\Omega}+(1-v_{i,j+d_{i,j}^{\Omega}}^{d,\Omega})\bar{v}_{i,j}^{R,\Omega}\}]. \tag{2}$$

The construction of the energy function $U_{edge}^{R,\Omega}$ is in a manner similar to that described for the energy function $U_{edge}^{L,\Omega}$.

### 4.3. *The matching block*

We write

$$U_{match}^{\Omega}(d^{\Omega})=\gamma_1^{\Omega}\sum_i\sum_j\left(x_{i,j}^{L,\Omega}-\frac{1}{|\eta_{i,j}|}\sum_{\alpha,\beta\in\eta_{i,j}}x_{\alpha,\beta+d_{i,j}^{\Omega}}^{R,\Omega}\right)^2(h_{i,j}^{L,\Omega}\oplus h_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})(v_{i,j}^{L,\Omega}\oplus v_{i,j+d_{i,j}^{\Omega}}^{R,\Omega})$$
$$+\gamma_2^{\Omega}\sum_i\sum_j\left\{\sum_{(p,q)\in W_{i,j}^{\Omega}}(d_{i,j}^{\Omega}-d_{p,q}^{\Omega})^2\left[\frac{1}{|p-i|+|q-j|}\left(\prod_{l=0}^{|i-p|}\prod_{m=0}^{|j-q|-1}(1-h_{\max(p,i)-l,\max(q,j)-m}^{d,\Omega})\right.\right.\right.$$
$$\left.\left.\left.\times|p-i|+\prod_{l=0}^{|i-p|-1}\prod_{m=0}^{|j-q|}(1-v_{\max(p,i)-l,\max(q,j)-m}^{d,\Omega})|q-j|\right)\right]\right\}$$
$$+\gamma_3^{\Omega}\sum_{i,j}\left\{1-\sum_k\delta\{(j+d_{i,j}^{\Omega})-(k+d_{i,k}^{\Omega})\}\right\}^2, \tag{3}$$

where

$$\delta(a-b)=\begin{cases}1 & \text{if } a=b,\\ 0 & \text{if } a\neq b.\end{cases} \tag{4}$$

The first term in (3) is the grey level matching at selected pixel locations, namely, at those locations where there is no vertical line field, or horizontal line field, or both vertical and horizontal line fields are not present. The reason for such a selection process is that, at edge points (i.e. points where $h_{i,j}=1$ or $v_{i,j}=1$), we cannot expect the grey levels to be similar in the left image and the right image. On the other hand, wherever there is no edge we can expect similar grey levels in the left image and the right image.

Next, since it is not very meaningful to compare one pixel in the left image with one pixel in the right image, we average the grey values around the pixel displaced by $d_{i,j}^{\Omega}$ in the neighborhood $\eta_{i,j}$ in the right image. Here $|\eta_{i,j}|$ represents the number of pixels over which the average is obtained. The second term is the smoothing term with the expression in the curly bracket as the controlling parameter. The disparity $d_{p,q}^{\Omega},(p,q)\in W_{i,j}^{\Omega}$ [$W_{i,j}^{\Omega}$ is a window around the pixel $(i,j)$], does not vary much provided there are no edges in the interpolated scene within the window $W_{i,j}^{\Omega}$. The term in the square bracket does precisely this by switching off smoothing

Vol. 331B, No. 5, pp. 531–554, 1994

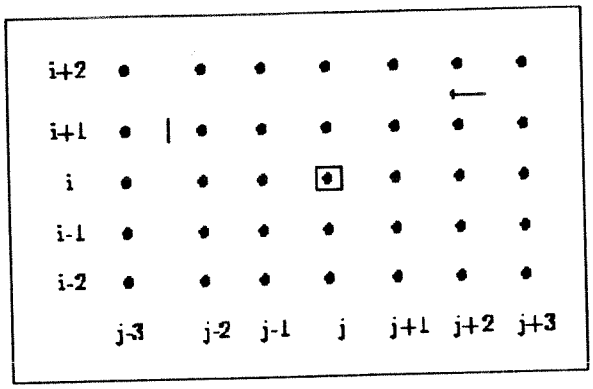


FIG. 3. A sample lattice.

within the window $W^{\Omega}_{i,j}$ whenever a line field is encountered in the interpolated image in the window $W^{\Omega}_{i,j}$. This term is better explained by looking at Fig. 3.

Consider a $7\times5$ window around the pixel $(i,j)$ in Fig. 3. Let the lines marked between the pixels represent the edge in the interpolated scene or the disparity map. Clearly the disparity value at $(i+1,j+2)$ and $(i+2,j+2)$ or $(i+1,j+2)$ and $(i,j)$ cannot be similar, because of the presence of an edge. In the same way the disparity value at $(i+1,j-3)$ and $(i+1,j-2)$ or $(i+1,j-3)$ and $(i,j)$ cannot be similar. Hence the disparity values should not be smoothed, and this is precisely being done by the term in the square bracket. The term in the square bracket returns a non-zero value only when there is no edge in the window under consideration.

Next we consider the incorporation of the two constraints, smoothness and uniqueness first proposed by Marr (**25**). In our case, Marr's smoothness constraint is modified to: *Disparity varies smoothly between edges.* The term in the curly bracket takes care of this; we refer to this as the selective smoothing term. The third term takes care of the uniqueness constraint, the idea being that each point in the left image should have one and only one corresponding point in the right image. More explicitly, it means that along any row $i$, if we were to calculate the disparity at the $j$th and the $k$th column (call it $d^{\Omega}_{i,j}, d^{\Omega}_{i,k}$), then according to the uniqueness constraint $j+d^{\Omega}_{i,j}\neq k+d^{\Omega}_{i,k}$; for the whole image this translates into the third term of (3). To our knowledge this energy function for the matching block is new.

### 4.4. *The interpolation block*

Write

$$U^{\Omega}_{interpol}(v^{d,\Omega},h^{d,\Omega})=\sum_i\sum_j\alpha^{\Omega}_1\{(d^{\Omega}_{i,j}-d^{\Omega}_{i-1,j})^2(1-h^{d,\Omega}_{i,j})+(d^{\Omega}_{i,j}-d^{\Omega}_{i,j-1})^2(1-v^{d,\Omega}_{i,j})\}$$

$$+\alpha^{\Omega}_2\{h^{d,\Omega}_{i,j}+v^{d,\Omega}_{i,j}\}+\alpha^{\Omega}_3\{(1-h^{L,\Omega}_{i,j})h^{d,\Omega}_{i,j}+(1-h^{d,\Omega}_{i,j})h^{L,\Omega}_{i,j}+(1-v^{L,\Omega}_{i,j})v^{d,\Omega}_{i,j}$$

$$+(1-v^{d,\Omega}_{i,j})v^{L,\Omega}_{i,j}\}+\alpha^{\Omega}_4\{(1-h^{R,\Omega}_{i,j+d^{\Omega}_{i,j}})h^{d,\Omega}_{i,j}+(1-h^{d,\Omega}_{i,j})h^{R,\Omega}_{i,j+d^{\Omega}_{i,j}}+(1-v^{R,\Omega}_{i,j+d^{\Omega}_{i,j}})v^{d,\Omega}_{i,j}$$

$$+(1-v^{d,\Omega}_{i,j})v^{R,\Omega}_{i,j+d^{\Omega}_{i,j}}\}. \tag{5}$$

The various terms in $U^{\Omega}_{interpol}$ can be explained in a manner similar to that

described for $U_{edge}^{L,\Omega}$ and $U_{edge}^{R,\Omega}$. The terms corresponding to $\alpha_1^\Omega$ and $\alpha_2^\Omega$ are respectively the *discontinuous regularization* term and the *line field penalty* term. The terms associated with $\alpha_3^\Omega$ and $\alpha_4^\Omega$ are the interaction terms which are driven by the output of the other modules.

## *V. Obtaining Images at Different Resolutions*

The energy functions constructed in Section IV are valid at all resolutions. In this section we sketch the method of Burt and Adelson (**20**) which obtains images at different resolutions.

Recall that $x_{i,j}^{R,\Omega}$ is the intensity value of the $(i,j)$th pixel in the right image at resolution $\Omega$ and $x_{i,j}^{R,\Omega-1}$ is the intensity value of the $(i,j)$th pixel at resolution $(\Omega-1)$, which is obtained using the algorithm proposed in (**20**). Note that for $x_{i,j}^{R,\Omega}$, $i$ and $j$ can take values in the range $(0, 2^{\Omega-1})$. Similarly, $x_{i,j}^{L,\Omega}$ and $x_{i,j}^{L,(\Omega-1)}$ represent the intensity of the left image at resolution $\Omega$ and $(\Omega-1)$ respectively.

The basic idea in the approach of (**20**) is as follows. An image at a given resolution is lowpass filtered so that high spatial frequencies are removed; as a result we can sample it at a lower rate (typically one half) and hence we have an image of lower resolution and one half the size of the original image. This process of lowpass filtering and subsampling results in images which are at different resolutions. In addition, the difference images at different resolutions are obtained by upsampling the coarser image by a factor of two, interpolating it, and then subtracting it from the next finer resolution image. The difference images so obtained are used by our algorithm to obtain the precomputed line fields at each resolution.

A suitable kernel for lowpass filtering is used to obtain images at different resolutions. If we assume a 1D signal and the size of the kernel to be 5, then as shown by Burt and Adelson (**20**) the weights of the kernel, denoted by $w(-2)$, $w(-1)$, $w(0)$, $w(1)$, $w(2)$, should satisfy the following constraints:

Normalization: $\sum_{i=-2}^{i=2} w(i) = 1$;

Symmetry: $w(i) = w(-i)$ for $i = 0, 1, 2$;

Equal Contribution: if $w(0) = a$, $w(1) = b$, $w(2) = c$, then $a+2c = 2b$ must be satisfied.

A $5 \times 5$ kernel

$$K = \begin{array}{|c|c|c|c|c|} \hline c^2 & bc & ac & bc & c^2 \\ \hline bc & b^2 & ab & b^2 & bc \\ \hline ac & ab & a^2 & ab & ac \\ \hline bc & b^2 & ab & b^2 & bc \\ \hline c^2 & bc & ac & bc & c^2 \\ \hline \end{array}$$

from (**20**) with $a = 0.4$, $b = 0.25$ and $c = 0.05$, was used to convolve with the high resolution ($\Omega$) image; the convolved image is then downsampled (for example,

Vol. 331B, No. 5, pp. 531–554, 1994

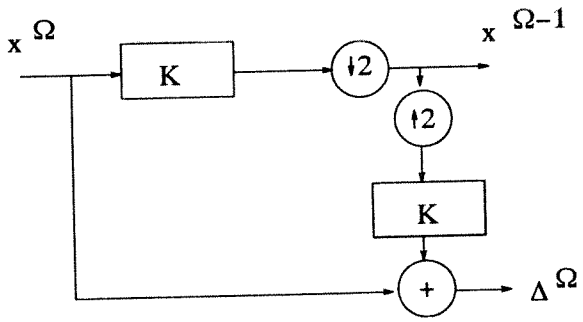


FIG. 4. Procedure for obtaining multiresolution images.

selecting every alternate pixel along each row and column) to obtain the image at lower resolution $(\Omega-1)$. The process of obtaining the low resolution image and the difference image is depicted in Fig. 4, where the block K represents convolution with the kernel K above, (↓2) represents downsampling by 2, and (↑2) represents upsampling by 2 (namely introducing a zero between alternate pixels). Figure 5 gives one such sequence of multiresolution images, while Fig. 6 depicts the corresponding difference image $\{\Delta^{\Omega}_{i,j}\}$.

### *VI. The Integrated Stereo Algorithm*

All four energy functions $U^{L,\Omega}_{edge}$, $U^{R,\Omega}_{edge}$, $U^{\Omega}_{match}$ and $U^{\Omega}_{interpol}$ need to be minimized. Minimization can be carried out using any known global minimization technique; in this paper we use the simulated annealing algorithm (**26**). The general structure of the integrated stereo algorithm is as follows:

Step 1: Use any standard edge detecting algorithm for obtaining the pre-processed line fields. We have used the difference image $\Delta^{\Omega}_{i,j}$ for this task since it provides a multiresolution framework, which will be used later on. Given the difference image $\Delta^{\Omega}_{i,j}$, line fields $\bar{v}^{\Omega}_{i,j}$ and $\bar{h}^{\Omega}_{i,j}$ are obtained by

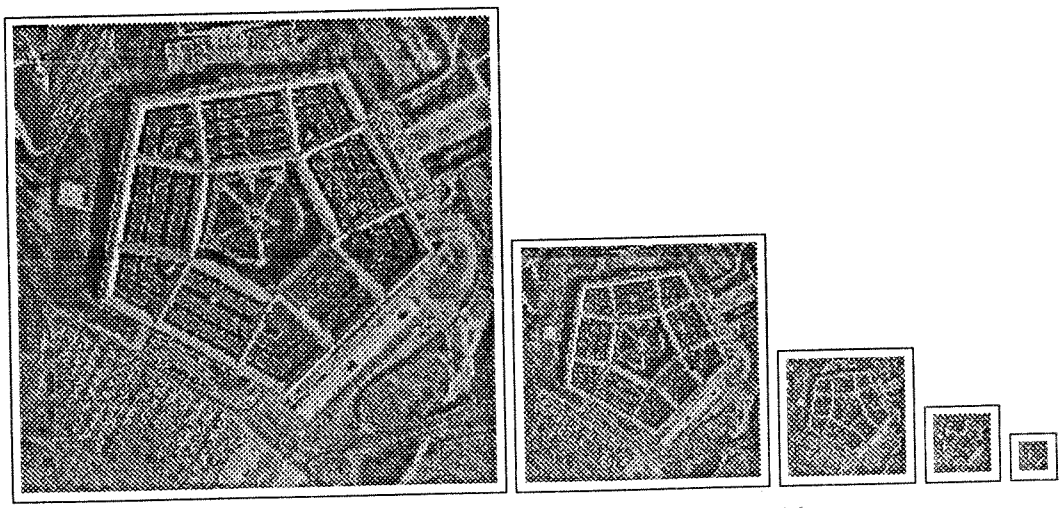

FIG. 5. Image at resolution 256, 128, 64, 32, 16.

$$\bar{v}^{\Omega}_{i,j} = \bar{h}^{\Omega}_{i,j} = \begin{cases} 1 & \text{if } |\Delta^{\Omega}_{i,j}| \geq \text{threshold} \\ 0 & \text{if } \Delta^{\Omega}_{i,j}| < \text{threshold.} \end{cases} \tag{6}$$

A superscript $L$ or $R$ to the line fields will indicate the precomputed line fields for the left or the right image. This becomes a fixed input for Steps 3–6.

Step 2: Initialize all the other line fields $v^{L,\Omega}$, $h^{L,\Omega}$, $v^{R,\Omega}$, $h^{R,\Omega}$, $v^{d,\Omega}$, $h^{d,\Omega}$ and the disparity $d^{\Omega}$ to zero.

Step 3: Using the values for $v^{L,\Omega}$, $h^{L,\Omega}$, $v^{R,\Omega}$, $h^{R,\Omega}$, $v^{d,\Omega}$, $h^{d,\Omega}$, $d^{\Omega}$ from the previous iteration and $x^{L,\Omega}$, $U^{L,\Omega}_{edge}$ is minimized. This outputs $v^{L,\Omega}$, $h^{L,\Omega}$.

Step 4: Using the results of Step 3, the values for $v^{R,\Omega}$, $h^{R,\Omega}$, $v^{d,\Omega}$, $h^{d,\Omega}$, $d^{\Omega}$ from the previous iterations and $x^{R,\Omega}$, $U^{R,\Omega}_{edge}$ is minimized. This outputs $v^{R,\Omega}$, $h^{R,\Omega}$.

Step 5: Using the results of Steps 3 and 4, the values of $v^{d,\Omega}$, $h^{d,\Omega}$, $d^{\Omega}$ from the previous iterations, $x^{R,\Omega}$ and $x^{L,\Omega}$, $U^{\Omega}_{match}$ is minimized. This outputs $d^{\Omega}$.

Step 6: Using the results of Steps 3, 4 and 5, and the values for $v^{d,\Omega}$, $h^{d,\Omega}$ from the previous iterations, $U^{\Omega}_{interpol}$ is minimized. This outputs $v^{d,\Omega}$, $h^{d,\Omega}$.

Step 7: Go to Step 3 if the stopping criterion is not satisfied. In all our simulations a fixed number of iterations determined the stopping criterion.

We refer to Steps 3–6 as one integrated iteration.

It is to be noted that the precomputed line fields $\bar{v}^{L,\Omega}$, $\bar{h}^{L,\Omega}$, $\bar{v}^{R,\Omega}$, $\bar{h}^{R,\Omega}$ are not updated though used in the integration model, the idea being that they are adequate enough and can be used by the model as a guiding line field map.

## *VII. Experiments*

The primary objective of the results presented here is to investigate the enhanced performance when various modules are integrated. To this end, we experiment with different variations derived from the general stereo integration model of

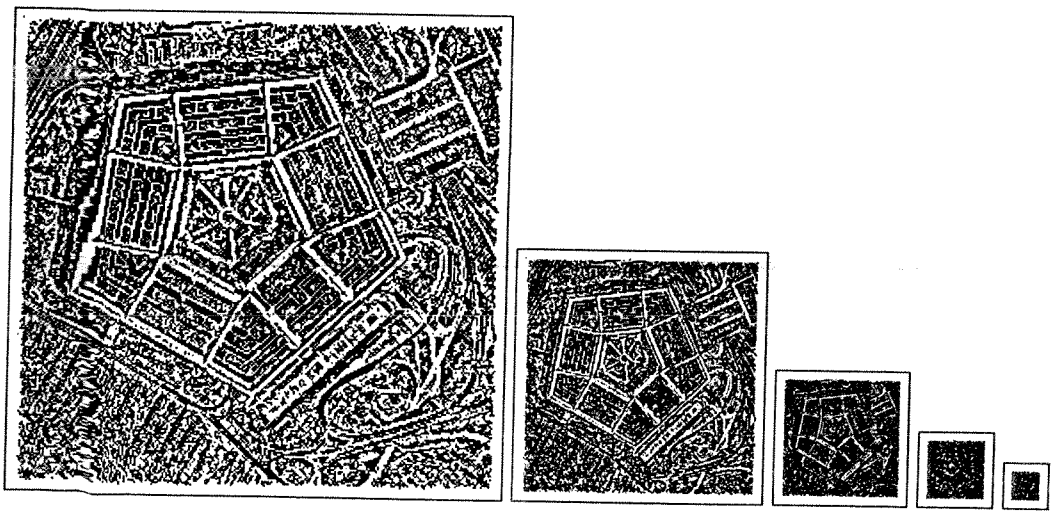

FIG. 6. Difference image at resolution 256, 128, 64, 32, 16.

Vol. 331B, No. 5, pp. 531–554, 1994

Fig. 2. Moreover, we also perform experiments to explore the advantages of multiresolution.

7.1. *Monoresolution (using only the finest resolution)*

Here we experiment with four different cases.

7.1.1. *No integration.* In this case we have the following one way structure: left and right image → left and right edge detector (precomputed) → matching module → interpolation. There is no feedback from any modules. We have used the difference image $\Delta_{i,j}$, for the highest resolution, to compute the right and the left edges $(\bar{h}, \bar{v})$ using (6).

7.1.2. *Only precomputed edges (Fig. 7).* In this integration model the precomputed edge module has feedforward (weak) interaction with the matching module as well as the interpolation module. On the other hand there is strong interaction between the matching module and the interpolation module. We use the difference image $\Delta^{\Omega}$ to calculate the precomputed line fields [see (6)] because the difference field is obtained as a byproduct while obtaining images at different resolutions, and also because computing line fields from the difference field is computationally simple unlike other edge detectors, for example the Canny edge detector.

7.1.3. *Interactive edge computation (Fig. 8).* In this model there is strong interaction between all the modules. The line fields are computed by minimizing $U_{edge}^{L,\Omega}$, $U_{edge}^{R,\Omega}$ (Sections 4.1 and 4.2).

7.1.4. *Precomputed edges and interactive edge computation.* This is the general integration model depicted in Fig. 2, and explained in Section II.

7.2. *Multiresolution (from coarse to fine)*

The integration process is similar to that carried out in the monoresolution case, except that we start at the coarsest resolution, and pass the disparity map to the next finer resolution as, for $0 \leqslant (i, j) \leqslant 2^{\Omega-1}-1$,

$$d^{\Omega}_{2\times i,2\times j} = 2 \times d^{\Omega-1}_{i,j},$$

$$d^{\Omega}_{2\times i+1,2\times j+1} = 0. \tag{7}$$

The multiplicative factor 2 corresponds to upsampling by 2. There are two main advantages of the multiresolution approach. The first is that we have a better initial disparity estimate at the finest resolution, and the second is significantly faster computation: let $I$ be the number of iterations required in the absence of multiresolution and $\mathcal{O}^{int\_only}(I2^{2\Omega})$ be the computational complexity of the integration algorithm. If $i$ is the number of iterations required at each resolution then the computational complexity of the algorithm which involves multiresolution is $\mathcal{O}^{int+multi}(i(2^{2\Omega}+2^{2(\Omega-1)}+2^{2(\Omega-2)}+\cdots+2^{2(\Omega-(\Omega-\omega))}))$ where $\omega$ is the coarsest resolution. Using the geometric series sum we have the order of complexity as $\mathcal{O}^{int+multi}(i2^{2\Omega}(1-(1/4)^{(\Omega-\omega+1)})/(1-(1/4)))$. Putting typical values of $I = 100$, $i = 10$, $\Omega = 8$, $\omega = 3$, we have $\mathcal{O}^{int+multi}/\mathcal{O}^{int\_only} \approx 1/7.5$.

We experiment with two cases in the multiresolution framework.

7.2.1. *Only precomputed edges.* The integration model is the same as that depicted

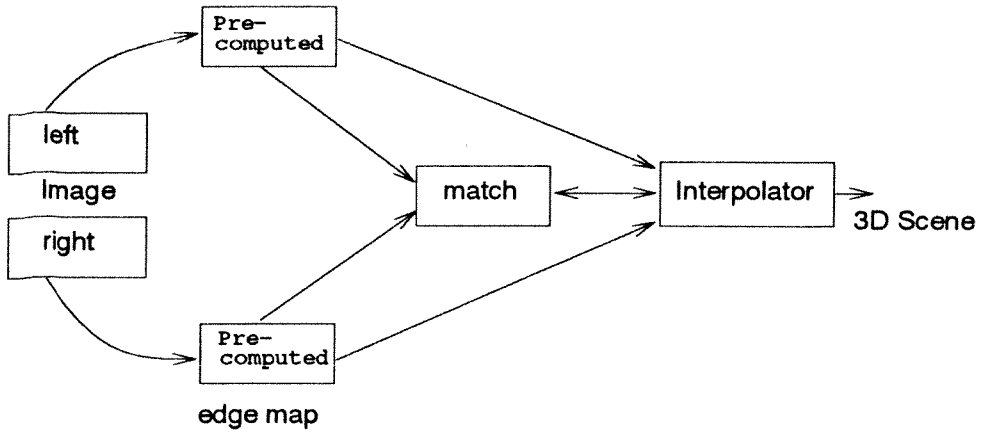


FIG. 7. The precomputed edge integration model.

in Fig. 7 except that it is executed for each resolution $\Omega$. We compute $\bar{h}^{\Omega}$ and $\bar{v}^{\Omega}$ using the difference image $\Delta_{i,j}^{\Omega}$ [Fig. 6, and (6)]. The image at different resolutions (Fig. 5) is input to the matching module.

7.2.2. *Interactive edge computation.* Here the integration model is as depicted in Fig. 8, except that it is executed for each resolution $\Omega$. Only the images at different resolutions (Fig. 5) are used as the input to the algorithm. The difference image is not used because the precomputed module is absent.

### 7.3. *Results*

Simulations were carried out to validate and justify the use of the proposed algorithm for obtaining surface description from binocular stereo. One of the main objectives was to show the enhanced performance when various modules are integrated. In addition, we wanted to study the effect of integration when a precomputed edge map was used to drive the integration model. We also wanted to explore the accuracy of the disparity map and the reduction in computation when the multiresolution approach was used in the stereo algorithm.

Simulations were carried out on three different pairs of stereo images of size 256. The first pair is the random dot stereogram (Figs 9 and 10); the two images are identical except for the fact that a small square portion of the left image (Fig. 9) has been shifted by ten pixels to the right and placed in Fig. 10. The second set (Figs. 11 and 12) is that of an auto part and the third pair (Figs. 13 and 14) is an aerial view of the Pentagon. In all our simulations we make use of the knowledge of the maximum disparity range in each of the stereo pair as shown in Table I.

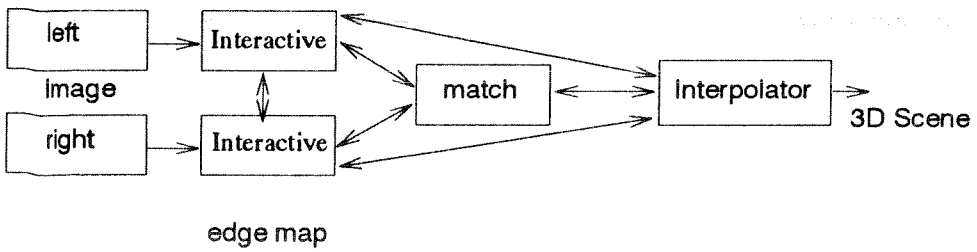


FIG. 8. The interactive edge integration model.

 

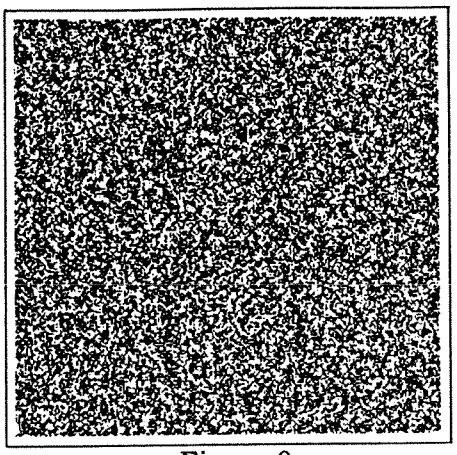

Figure 9

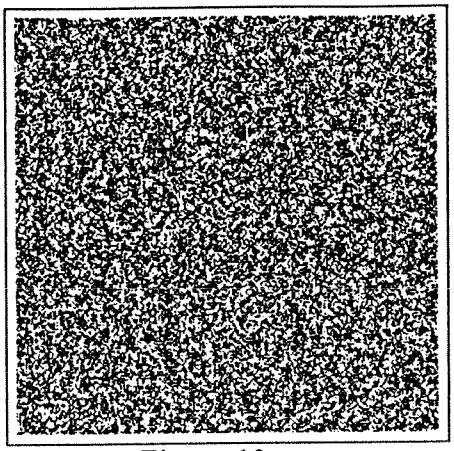

Figure 10

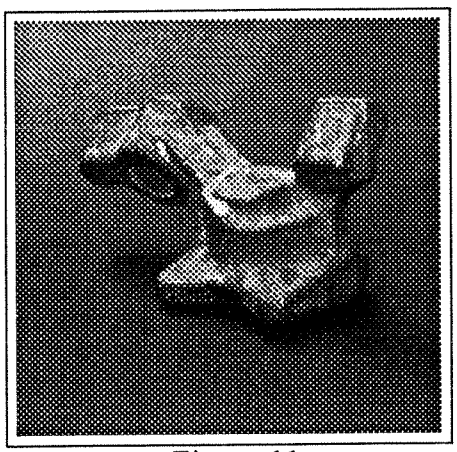

Figure 11

Figure 12

Figure 13

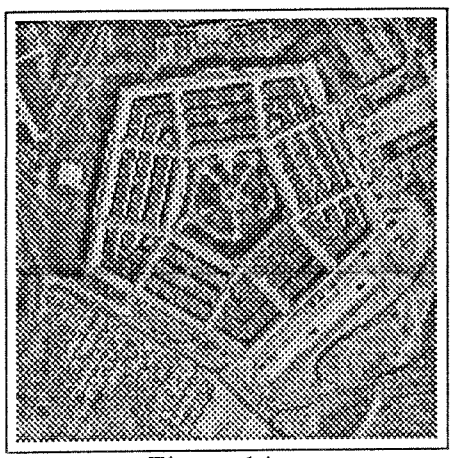

Figure 14

TABLE I.
*Disparity range*

| Image | Disparity range |
|---|---|
| Random dot stereogram | 10 |
| Auto part | 36 |
| Pentagon | 5 |

Figures 15–26 are concerned with the random dot stereogram (Figs 9 and 10). The first set of results (Figs 15 and 16) are obtained when no integration is used (Section 7.1.1), Fig. 15 is the disparity plotted as an intensity map (black shows zero disparity) and Fig. 16 is the disparity plotted as a depth map. The next set of results (Figs 17 and 18) are the intensity plot and the depth map of the disparity using the precomputed edge integration model of Fig. 7. Figures 19 and 20 are obtained using the interactive edge integration model of Fig. 8. Figure 21 is the intensity plot of the disparity and Fig. 22 is the depth plot of the disparity using the general stereo integration model of Fig. 2. Figures 23–26 are obtained using the multiresolution technique. Figures 23 and 24 are the disparity plotted as an intensity map and disparity plotted as depth for the case where only precomputed edges are used along with multiresolution (Section 7.2.1). Figures 25 and 26 are obtained using interactive edge computation along with multiresolution (Section 7.2.2).

Figures 27–38 and Figs 39–50 depict the results of various experiments using the auto part stereo pair and the Pentagon stereo pair, respectively. The sequence of operations is the same as that depicted for the random dot stereogram and this follows the sequence of subsections and sub-subsections of Section VII. This sequence is: no integration, only precomputed edges, interactive edge computation, precomputed edges and interactive edge computation, only precomputed edged using multiresolution, and interactive edges using multiresolution. In each case the intensity disparity map as well as the 3D display of the disparity map are shown.

In all our simulations the number of integrated iterations was fixed at 200 for the monoresolution case and 20 for the multiresolution case. The value of the threshold in (6) was set to 20.

The parameter values are shown in Table II. Those that are not shown (i.e. $\lambda_1$, $\beta_1$, $\alpha_1$, $\gamma_1$) take the value 1. The value of the parameters in the case of multiresolution was kept identical over all resolutions.

7.3.1. *Observations.*

1. There is a significant improvement in the disparity map obtained when integration is used, compared to when no integration is used (compare Fig. 16 with Figs 18, 20, 22, 24 and 26, and compare Fig. 28 with Figs 30, 32, 34, 36 and 38 respectively).
2. Based on the simulations for the random dot stereogram we see that the accuracy of the disparity map is better when we use the multiresolution framework (Sections 7.2.1 and 7.2.2). Nevertheless, for the case of auto part

Vol. 331B, No. 5. pp. 531–554, 1994

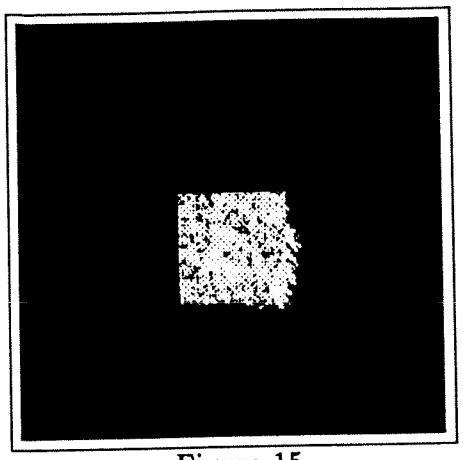

Figure 15

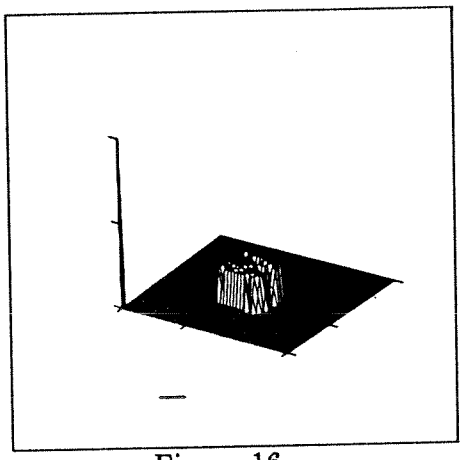

Figure 16

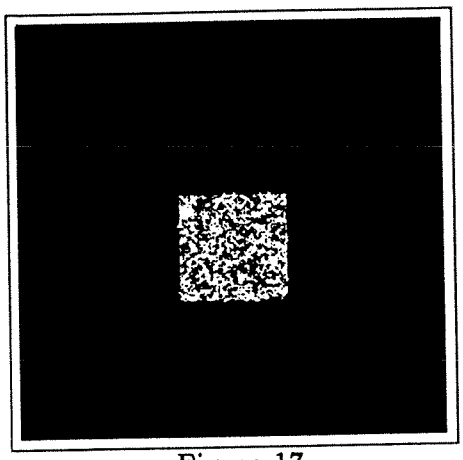

Figure 17

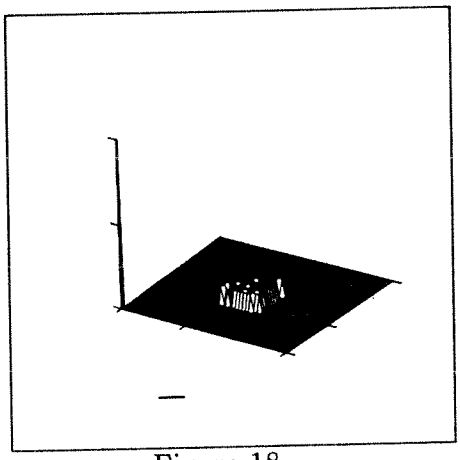

Figure 18

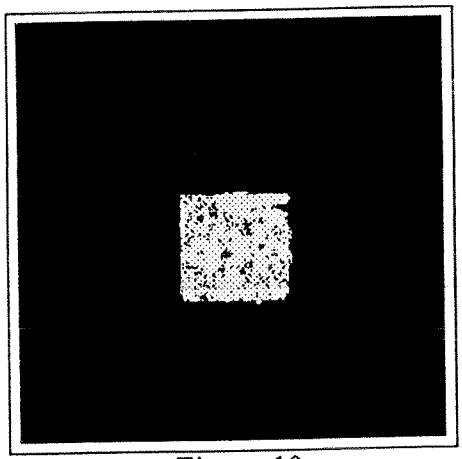

Figure 19

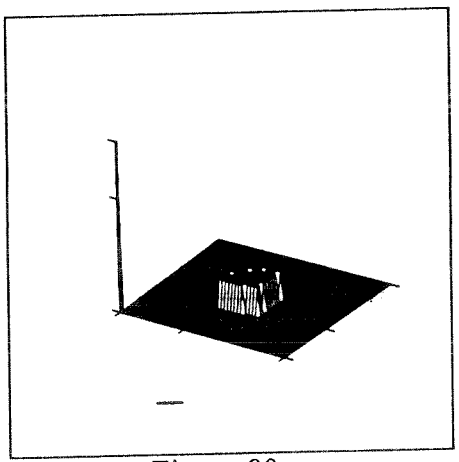

Figure 20

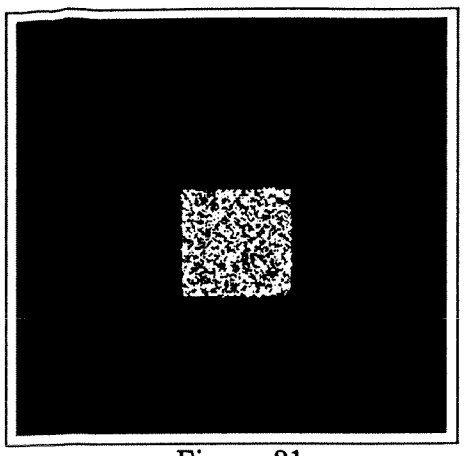

Figure 21

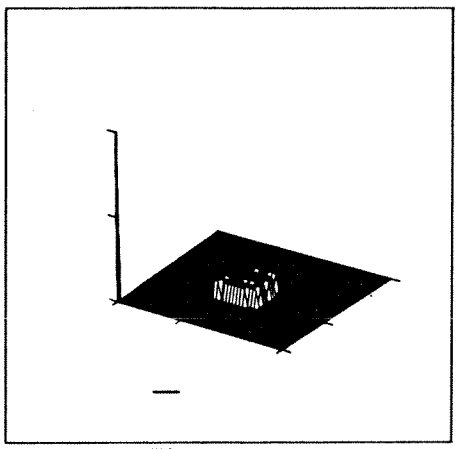

Figure 22

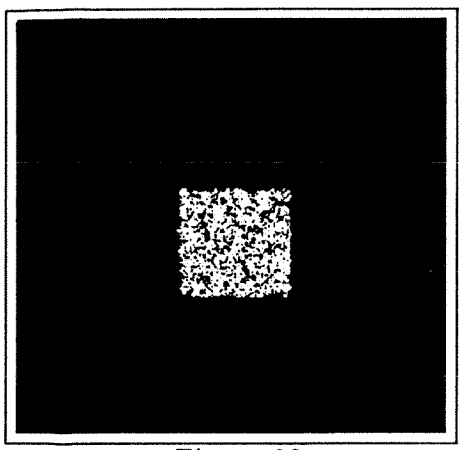

Figure 23

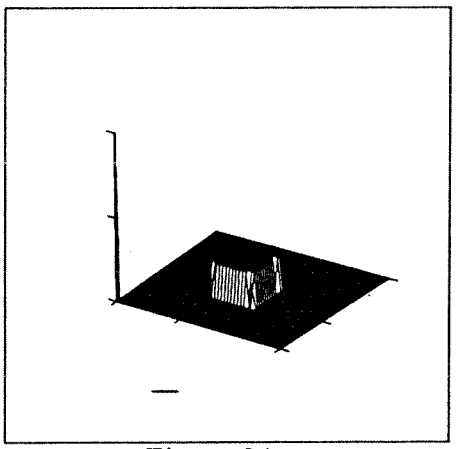

Figure 24

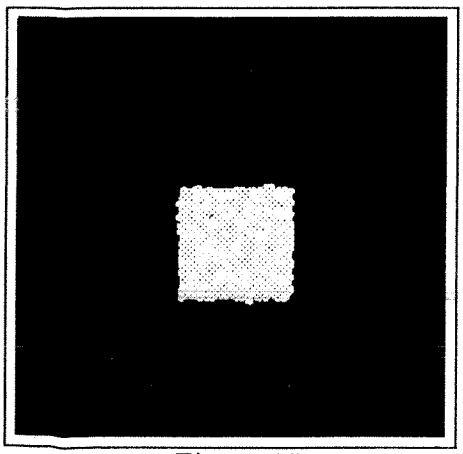

Figure 25

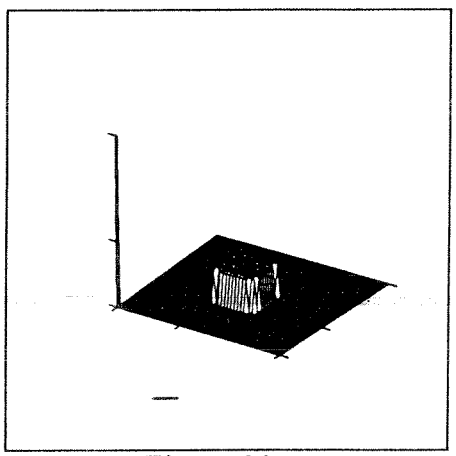

Figure 26

Vol. 331B, No. 5, pp. 531–554, 1994
Printed in Great Britain. 

Figure 27

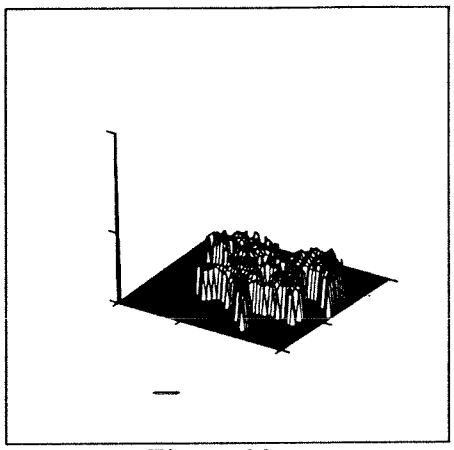

Figure 28

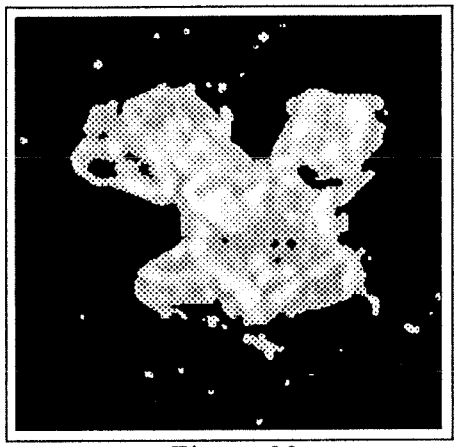

Figure 29

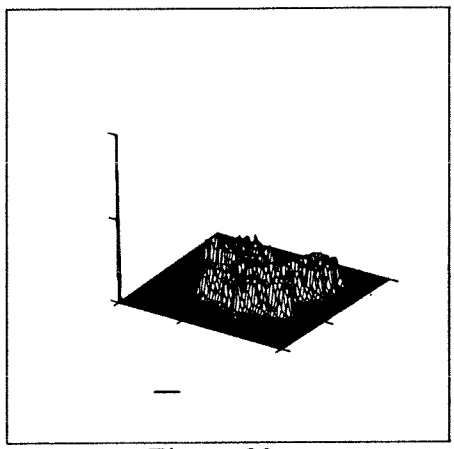

Figure 30

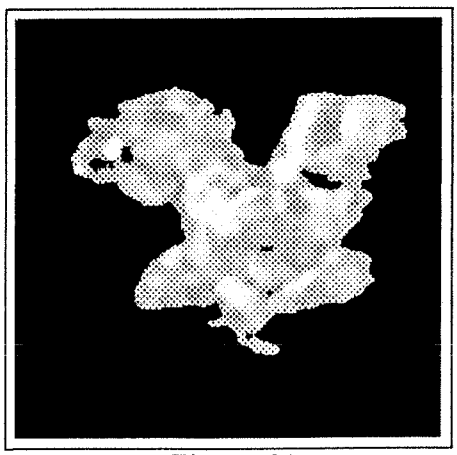

Figure 31

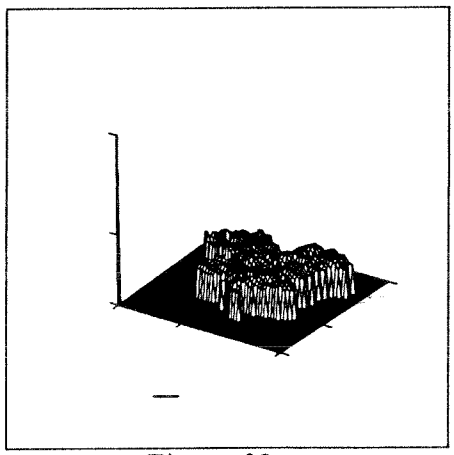

Figure 32

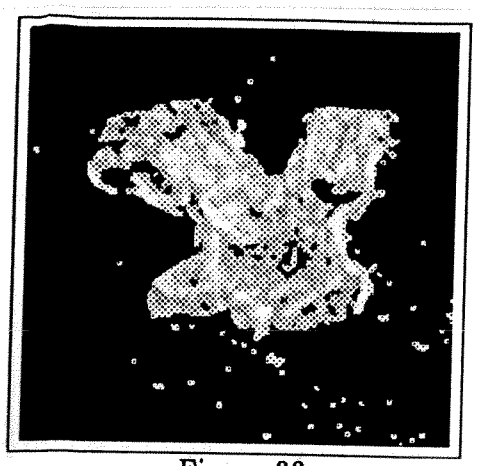

Figure 33

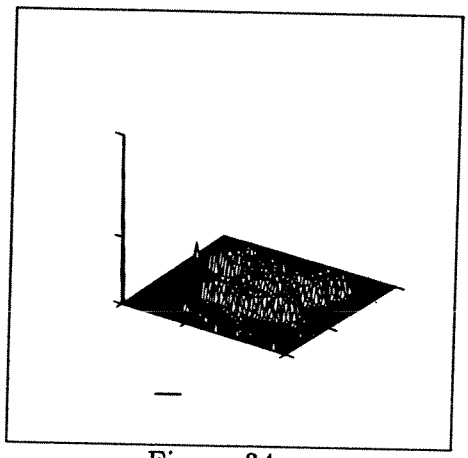

Figure 34

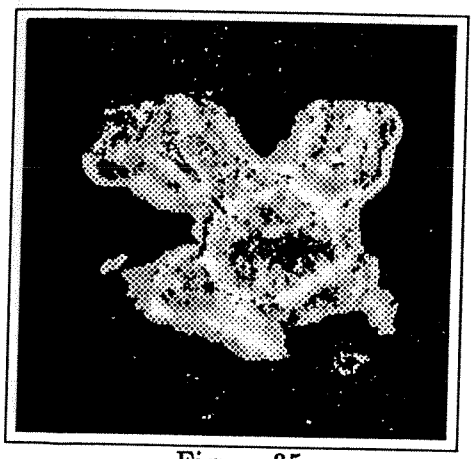

Figure 35

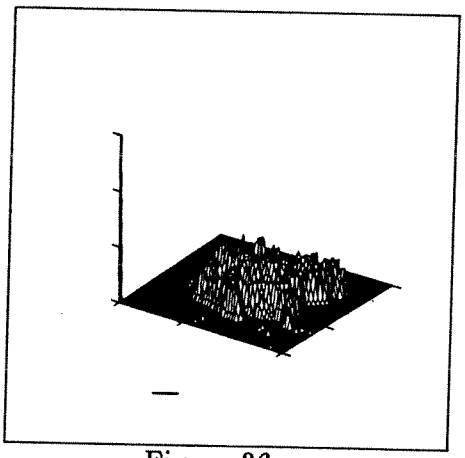

Figure 36

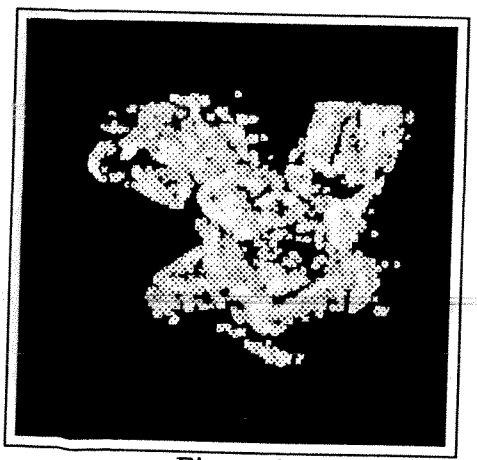

Figure 37

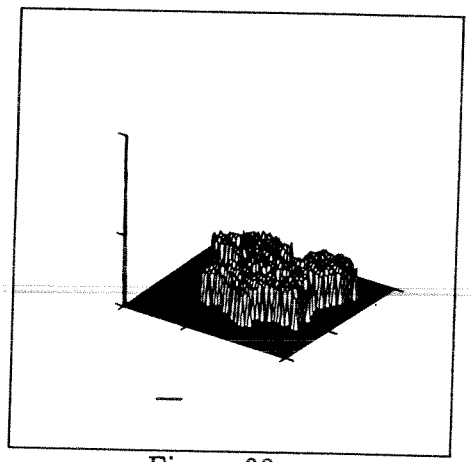

Figure 38

Vol. 331B, No. 5, pp. 531–554, 1994

TABLE II.
*Values of parameters used in simulation*

| | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\lambda_7$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $\beta_5$ | $\beta_6$ | $\beta_7$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | $\gamma_2$ | $\gamma_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Monoresolution | | | | | | | | | | | | | | | | | |
| Precomputed | 100 | 10 | 10 | 0 | 0 | 0 | 100 | 10 | 10 | 0 | 0 | 0 | 100 | 10 | 10 | 50 | 1 |
| Interactive | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 1 |
| Precomputed Interactive | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 1 |
| Multiresolution | | | | | | | | | | | | | | | | | |
| Precomputed | 100 | 10 | 10 | 0 | 0 | 0 | 100 | 10 | 10 | 0 | 0 | 0 | 100 | 10 | 10 | 50 | 1 |
| Interactive | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 10 | 10 | 150 | 10 | 10 | 10 | 1 |

and the Pentagon, much better results are obtained using the interactive line field computation model (Section 7.1.3) or the multiresolution interactive line field computation model (Section 7.2.2) (see Figs 32 and 38, or Figs 44 and 50).

3. The interactive edge computation model seems to work better when compared with the precomputed edge model (compare Fig. 31 with Fig. 29, and compare Fig. 43 with Fig. 41).
4. The multiresolution interactive edge computation model is computationally less expensive (see Table III). One sees that for the simulations, the multiresolution approach is faster by a factor of eight.
5. When both precomputed line fields and interactively computed line fields are used one sees that the results are better than those for the case of only precomputed line fields (Section 7.1.2), but worse than those for only interactive line field computed (Section 7.1.3). Compare Fig. 33 with Figs 29 and 31.

   The reason for this could be that the precomputed line fields are dominating the integration process, i.e. the model is driven more by the precomputed line fields and less by the interactively computed line fields.

TABLE III.
*Computational time to get final results* (*on* PC-486)

| Model | Time taken (in seconds) |
|---|---|
| Monoresolution | |
| No integration | 7700 |
| Precomputed edge | 8400 |
| Interactive edge | 9800 |
| Precomputed and interactive edges | 10000 |
| Multiresolution | |
| Precomputed edge | 1000 |
| Interactive edge | 1190 |

6. Based on the simulations, one sees that the multiresolution interactive edge computation integration model (Section 7.2.2) gives the best results in terms of computation time and accuracy of the disparity map.

## *VIII. Conclusions*

A new integrated stereo vision algorithm for estimating a dense disparity map has been formulated and implemented, using the intensity as the only cue. The stereo integration scheme is investigated in a monoresolution and multiresolution framework.

We would like to remark that the way the algorithm has been formulated, it lends itself to being implemented in parallel, using a Hopfield-like neural network. Note that in all the energy modules, except the matching module, it is required that two binary fields (corresponding to the horizontal and vertical line fields) be updated, and this can be accomplished by using a Hopfield neural network. The matching block can also be put into the Hopfield neural network framework because the disparity is usually confined to a small range. For this purpose one could use a multilevel threshold function analogous to the multilevel sigmoidal function of (27).

## *References*

(1) S. T. Barnard and M. A. Fischer, "Computational stereo", *Comput. Surveys*, Vol. 14, pp. 553–572, December 1982.

(2) M. Bertero and T. A. Poggio, "Ill-posed problems in early vision", *IEEE Proc.*, Vol. 76, pp. 869–889, August 1988.

(3) N. M. Nasarabadi and C. Y. Choo, "Hopfield network for stereo vision correspondence", *IEEE Tran. Neural Net.*, pp. 5–13, January 1992.

(4) S. Pollard, J. Mayhew and J. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit", *Perception*, Vol. 14, pp. 449–470, 1985.

(5) D. Marr and T. Poggio, "A computational theory of human stereo vision", *Proc. Roy. Soc. London*, Vol. B-204, pp. 309–328, 1979.

(6) W. E. L. Grimson, "Computational experiments with feature based stereo algorithm", *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. 7, pp. 17–34, January 1985.

(7) K. Prazdny, "Detection of binocular disparities", *Biol. Cybern.*, Vol. 52, pp. 93–99, 1985.

(8) R. D. Eastman and A. M. Waxman, "Using disparity functionals for stereo correspondence and surface reconstruction", *Comp. Vis. Graph. and Image Processing*, Vol. 39, pp. 73–101, 1987.

(9) S. T. Barnard and W. B. Thompson, "Disparity analysis of images", *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. 2, pp. 330–340, February 1980.

(10) Y. C. Kim and J. K. Agarwal, "Positioning 3D objects using stereo images", *IEEE J. Robotics Autom.*, Vol. 3, pp. 361–373, August 1987.

(11) D. Marr and T. Poggio, "Cooperative computation of stereo disparity", *Science*, Vol. 194, pp. 283–287, 1976.

(12) E. B. Gamble, D. Geiger, T. Poggio and D. Weinshall, "Integration of vision modules and labeling of surface discontinuities", *IEEE Trans. Syst. Man & Cybern.*, Vol. 19, pp. 1576–1581, November/December 1989.

(13) S. T. Toborg and K. Hwang, "Cooperative vision integration through data-parallel neural computations", *IEEE Trans. Comput.*, Vol. 40, pp. 1368–1379, December 1990.
(14) W. Hoff and N. Ahuja, "Surfaces from stereo: Integration feature matching, disparity estimation and contour detection", *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. 11, pp. 121–136, February 1989.
(15) T. Kanade, M. Okutomi and T. Nakahara, "A multiple-baseline stereo method", "Proc. DARPA Image Understanding Workshop", pp. 409–426, January 1992.
(16) S. T. Barnard, "Stochastic stereo matching over scales", *Int. J. of Comp. Vis.*, No. 3, pp. 17–32, 1989.
(17) H. Isil Bozma and J. S. Duncan, "Integration of stereo modules: A game theoretic approach", "Proc. Int. Conf. on Comp. Vis.", pp. 501–507, 1991.
(18) J. J. Clark and A. L. Yullie, "Data Fusion for Sensory Information Processing Systems", Kluwer Academics Publishers, Netherlands, 1990.
(19) H. P. Moravec, "Towards automatic visual obstacle avoidance", "Proc. 5th Int. Conf. on Artificial Intell.", Cambridge, MA, pp. 584, 1977.
(20) P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Trans. on Comm.*, Vol. 31, pp. 532–540, April 1983.
(21) P. Perez and F. Heitz, "Multiscale MRF and constrained relaxation in low level image analysis", *ICASSP* 92, pp. III 61–64, 1992.
(22) D. De Vleeschauwer, "An intensity based, coarse to fine approach to reliable binocular disparity", *CVGIP: Image Understanding*, Vol. 57, pp. 204–218, March 1993.
(23) S. D. Cochran and G. Medoni, "3D surface description from binocular stereo", *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. 14, pp. 981–994, October 1992.
(24) S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images", *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. 6, pp. 721–741, 1984.
(25) D. Marr, "Vision", W. H. Freemaan and Co., San Francisco, 1982.
(26) E. Aarts and J. Korst, "Simulated Annealing and Boltzmann Machines", John Wiley, NY, 1989.
(27) K. Sivakumar and U. B. Desai, "Image restoration using a multilayer perceptron with a multilevel sigmoidal function", *IEEE Trans. Sig. Processing*, Vol. 41, pp. 2018–2022, May 1993.