A parallel stereo algorithm that produces dense depth maps and preserves image features

Pascal Fua

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1369

Programme 4
Robotique, Image et Vision

A PARALLEL STEREO ALGORITHM THAT PRODUCES DENSE DEPTH MAPS AND PRESERVES IMAGE FEATURES

Pascal FUA

Janvier 1991

RR_1369

A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features

Pascal Fua (fua@mirsa.inria.fr)
INRIA Sophia-Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex
France

Submitted to the journal of Machine Vision and Applications

Abstract

To compute reliable dense depth maps a stereo algorithm must preserve depth discontinuities and avoid gross errors. In this paper, we show how simple and parallel techniques can be combined to achieve this goal and deal with complex real world scenes.

Our algorithm relies on correlation followed by interpolation. During the correlation phase the two images play a symmetric role and we use a validity criterion for the matches that eliminates gross errors: at places where the images cannot be correlated reliably, due to lack of texture or occlusions for example, the algorithm does not produce wrong matches but a very sparse disparity map as opposed to a dense one when the correlation is successful. To generate dense depth map, the information is then propagated across the featureless areas but not across discontinuities by an interpolation scheme that takes image grey levels into account to preserve image features.

We show that our algorithm performs very well on difficult images such as faces and cluttered ground level scenes. Because all the algorithms described here are parallel and very regular they could be implemented in hardware and lead to extremely fast stereo systems.

Un algorithme stéréo parallèle: calcul de cartes de profondeur denses qui préservent les discontinuités

Pascal Fua (fua@mirsa.inria.fr)
INRIA Sophia-Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex
France

Soumis au journal Machine Vision and Applications

Abstract

Pour calculer des cartes de profondeur dense qui sont exactes, un algorithme stéréographique doit préserver les discontinuités de profondeur tout en évitant les erreurs grossières. Dans ce rapport, nous montrons comment combiner des techniques simples et parallèles pour obtenir ce résultat et traiter efficacement des images complexes.

Notre algorithme calcule une première carte de disparité par corrélation puis l'interpole pour produire une carte dense. Durant la première étape, les deux images jouent un rôle symétrique et nous utilisons un critère de validité qui nous permet de rejeter les erreurs grossières: aux endroits où les deux images ne peuvent être corrélées, à cause d'un manque de texture ou de la présence d'occlusions par exemple, l'algorithme ne construit pas de mises en correspondance erronées mais plutôt une carte peu dense par opposition à une carte presque dense ailleurs. Pour compléter la carte, nous propageons ensuite l'information à travers les zones non texturées tout en utilisant les niveaux de gris de l'image pour préserver les discontinuités.

Nous montrons que notre méthode nous permet d'obtenir d'excellents résultats pour des scènes complexes, visages et vues au ras du sol par exemple. Les algorithmes que nous utilisons sont tous réguliers et parallèles; leur implantation pourrait donc se faire sur des architectures spécialisées ce qui conduirait à un système extrêmement rapide.

1 Introduction

Over the years numerous algorithms for passive stereo have been proposed, they can roughly be classified in three categories [4]:

1. **Feature Matching.** Those algorithms extract features of interest from the images, such as edge segments or contours, and match them in two or more views. These methods are fast because only a small subset of the image pixels are used, but may fail if the chosen primitives cannot be reliably found in the images; furthermore they usually yield very sparse depth maps.

2. **Correlation.** In these approaches, the system attempts to correlate the grey levels of image patches in the views being considered, assuming that they present some similarity. This assumption appears to be a valid one for relatively textured areas; however if may prove wrong at occlusion boundaries and within featureless regions.

3. **Regularization.** The depth is computed by fitting a smooth depth map that accounts for the disparities between the two images. A dense depth map is produced, which is an advantage over the previous approaches; however the smoothness assumptions that are required may not always be satisfied.

Image	Ranking of [9]	Number of results	Standard deviation	Best	Worst
1	5	25	0.21	0.16	1.79
2	5	12	0.30	0.18	26.54
3	1	5	0.38	<-	1.67
4	3	11	0.81	0.73	29.51
5	1	2	0.34	<-	5.44
6	1	3	0.14	<-	6.10
7	3	15	0.27	0.20	2.78
8	1	3	0.32	<-	2.58
9	1	3	0.15	<-	6.41
10	1	14	0.26	<-	7.23
11	-	2	-	0.64	0.69
12	2	6	0.43	0.32	5.42

Table 1: Performance of various stereo systems [8]: The test data set was composed of twelve image pairs varying in resolution from 1:20 to 1:30000 and was sent to 15 research institutes across the world. For each image, we list the number of results received, the standard deviation between Hannah's results [9] and a manually generated disparity map (in pixels), as well as the standard deviations for the systems that performed best and worse.

All these techniques have their strengths and weaknesses and it is difficult to assess their compared merits since few researchers work on similar data sets. However, one can get a feel for the relative performance of these systems from the study by Güelch [8]. In this work, the author has assembled a standardized data set and sent it two 15 research institutes across the world. It appears that the correlation based system developed at SRI by Hannah [9] has produced the best results both in terms of precision and reliability. Unfortunately this system only matches a very small proportion, typically less than 1%, of the image points.

In this paper we propose a correlation algorithm that reliably produces far denser maps with very few false matches and can therefore be effectively interpolated. In the next section we describe our hypothesis generation mechanism that attempts to match every point in the image and uses a consistency criterion to reject invalid matches. This

criterion is designed so that when the correlation fails, instead of yielding an incorrect answer, the algorithm returns NO answer. As a result, the density of the computed disparity map is a very good measure of its reliability. The interpolation technique described in the section that follows combines the depth map produced by correlation and the grey level information present in the image itself to introduce depth discontinuities and fit a surface that is piecewise smooth. These algorithms have proven very effective on real data. Their parallel implementation on a Connection Machine[TM][1] relies only on local operations and on nearest neighbor communication; they could be ported to a dedicated architecture, thereby making fast and cheap systems possible.

2 Correlation

Most correlation based algorithms attempt to find interest points on which to perform the correlation. While this approach is justified when only limited computing resources are available, with modern hardware architectures and massively parallel computers it becomes possible to perform the correlation over the whole image and retain only results that appear to be "valid." The hard problem is then to provide an effective definition of what we call validity and we will propose one below.

In our approach, we compute correlation scores for every point in the image by taking a fixed window in the first image and a shifting window in the second. The second window is moved in the second image by integer increments along the epipolar line and an array of correlation scores is generated for integer disparity values. In this work we use normalized correlation of grey level values and take the correlation score s to be:

$$\begin{aligned} s &= \max(0, 1 - c) \\ c &= \frac{\sum_{i,j}((I_1(x+i,y+j) - \overline{I_1}) - (I_2(x+dx+i,y+dy+j) - \overline{I_2}))^2}{\sqrt{(\sum_{i,j}(I_1(x+i,y+j) - \overline{I_1})^2)(\sum_{i,j}(I_2(x+dx+i,y+dy+j) - \overline{I_2})^2)}} \end{aligned} \quad (1)$$

where I_1 and I_2 are the left and right image intensities, $\overline{I_1}, \overline{I_2}$ are their average value over the correlation window and dx, dy represent the displacement along the epipolar line. The measured disparity can then be taken to be the one that provides the highest value of s. In fact, to compute the disparity with subpixel accuracy, we fit a second degree curve to the correlation scores in the neighborhood of the maximum and compute the optimal disparity by interpolation.

We find the normalized correlation score useful because it is insensitive to linear transformation of the images which may result from slightly different settings of the cameras. An alternative would be to perform non normalized correlation, i.e. take c to be simply

$$c = \frac{\sum_{i,j}(I_1(x+i,y+j) - I_2(x+dx+i,y+dy+j))^2}{\sqrt{(\sum_{i,j} I_1(x+i,y+j)^2)(\sum_{i,j} I_2(x+dx+i,y+dy+j)^2)}} , \quad (2)$$

where I_1 and I_2 are difference of gaussians of the original images so that their mean within the correlation window is close to zero. This second method is computationally slightly more effective but the band-passing of the images discards some of the relevant texture and may lead to less accurate results in practice.

2.1 Validity of the Disparity Measure

As shown by Nishihara [15], the probability of a mismatch goes down as the size of the correlation window and the amount of texture increase. However, using large windows leads to a loss of accuracy and the possible loss of important image features. For smaller windows, the simplest definition of validity would call for a threshold on the correlation score; unfortunately such a threshold would be rather arbitrary and, in practice, hard to choose. Another approach is to build a correlation surface by computing disparity scores for a point in the neighborhood of a prospective match and checking that the surface is peaked enough [1]. It is more robust but also involves a

[1] TMC Inc.

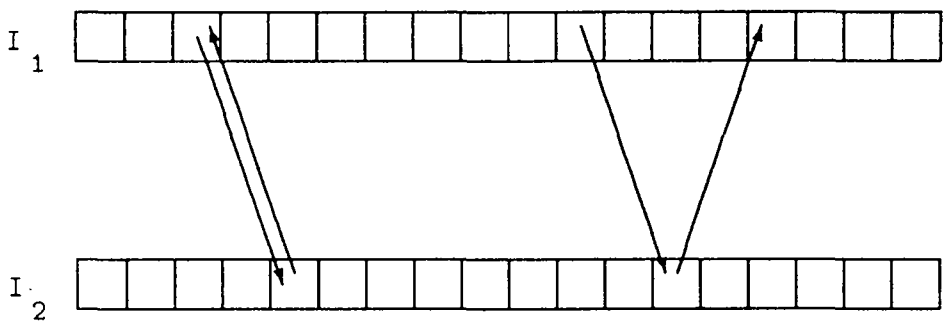


Figure 1: Consistent vs inconsistent matches: the two rows represent pixels along two epipolar lines of I_1 and I_2 and the arrows go from a point in one of the images towards the point in the other image that maximizes the correlation score. The match on the left is consistent because correlating from I_1 to I_2 and from I_2 to I_1 yields the same match unlike the matches on the right that are inconsistent.

set of relatively arbitrary thresholds. Here we propose a definition of a valid disparity measure in which the two images play a symmetric role and that allows us to reliably use small windows. We perform the correlation twice by reversing the roles of the two images and consider as valid only those matches for which we measure the same depth at corresponding points when matching from I_1 into I_2 and I_2 into I_1. As shown in Figure 1, this can be defined as follows.

> Given a point P_1 in I_1, let P_2 be the point of I_2 located on the epipolar line corresponding to P_1 such that the windows centered on P_1 and P_2 yield the optimal correlation measure. The match is valid if and only if P_1 is also the point that maximizes the score when correlating the window centered on P_2 with windows that shift along the epipolar line of I_1 corresponding to P_2.

(a) (b) (c)

Figure 2: (a) An outdoor scene with two trees and a stump. The same scene seen from the left (b) and the right (c) so that different parts of the ground are occluded by the trees.

For example, the validity test is likely to fail in presence of an occlusion. Let us assume that a portion of a scene is visible in I_1 but not I_2. The pixels in I_1 corresponding to the occluded area in I_2 will be matched, more or less

at random, to points of I_2 that correspond to different points of I_1 and are likely to be matched with them. The matches for the occluded points will therefore be declared invalid and rejected. We illustrate this behaviour using the portion of the tree scene of Figure 2 outlined in Figure 2(a).[2]. Different parts of the ground between the two trees and between the trees and the stump are occluded in Figures 2 (b) and (c). In Figures 3 (a) and (b), we show the computed disparities for this image window after correlation with the images shown in Figures 2(b) and 2(c) respectively. The points for which no valid match can be found appear in white and the areas where their density becomes very high correspond very closely to the occluded areas for both pairs of images. These results have been obtained using 3x3 correlation windows; these small windows are sufficient in this case because the scene is very textured and gives our validity test enough discriminating power to avoid errors. We will elaborate on this point in appendix A. For an image like this one it is a distinct advantage to be able to use small windows because the correct depth of the ground behind the trees could not be computed with larger ones that would include the tree trunks.

(a) (b) (c)

Figure 3: (a) The result of matching 2(a) and 2(b) for window of 2(a) delimited by the white rectangle. (b) The result of matching 2(a) and 2(c) for the same windows. (c) The merger of four disparity maps computed using the image of Figure 2 (a) as a reference frame, the two other images of 2 and two additional images. Invalid matches appear in white and become almost dense in occluded areas of (a) and (b). The closest areas are darker; note that they are few false matches although the correlation windows used in this case are very small (3x3).

We use the face shown in Figure 4 to demonstrate another case in which the validity test rejects false matches. The epipolar lines are horizontal and in Figure 4 (d) we show the resulting disparity image, using 7x7 windows, in which the invalid matches appear in black. In Figure 4 (e) we show another disparity image computed after having shifted one of the images vertically by two pixels, thereby degrading the calibration and the correlation. Note that the disparity map becomes much sparser but that no gross errors are introduced. In practice, we take advantage of this behaviour for poorly calibrated images: we compute several disparity maps by shifting one of the images up or down and retaining the same epipolar lines[3], thereby replacing the epipolar line by an epipolar band, and retain the valid matches with the highest correlation score.

In the two examples above, we have shown that when the correlation between the two images of a stereo pair is degraded our algorithms tends, instead of making mistakes, to yield sparse maps. This actually is a very generic behaviour that we further discuss below and in appendix A.

Generally speaking, correlation based algorithms rely on the fact that the same texture can be found at corresponding points in the two images of a stereo pair and are known to fail when:

[2] Courtesy of SRI International

[3] assumed not to be exactly vertical

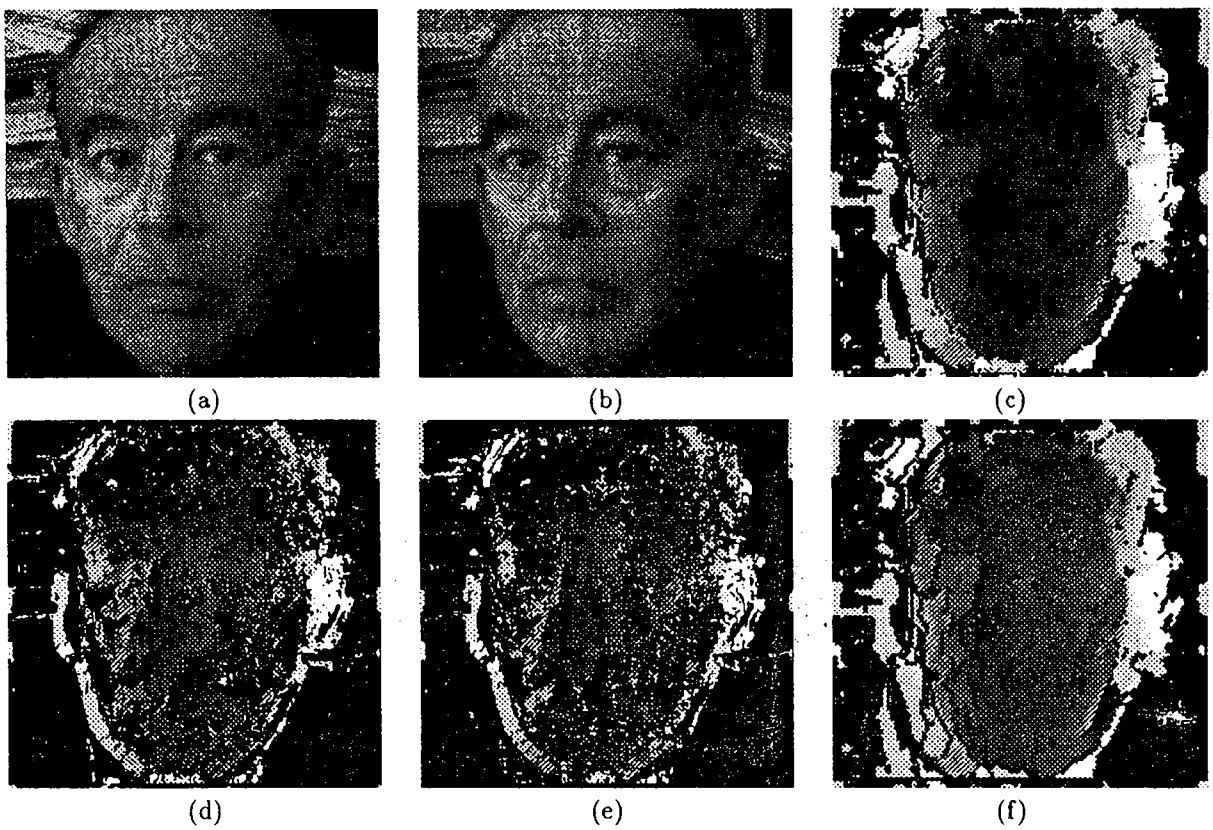


Figure 4: (a) (b) Left and right 256x256 images of a face. (c) The disparity map obtained by merging the results computed at two levels of resolution. (d) Disparity map computed at the highest resolution. (e) Disparity map computed at the highest resolution after shifting the right up by one pixel. (f) Disparity map computed at the lower resolution.

- The areas to be correlated have little texture.
- The disparities vary rapidly within the correlation window.
- There is an occlusion.

If we consider the local image texture as a signal to be found in both images, we can model these problems as noise that corrupts the signal. In appendix A, we use synthetic data to show that as the noise to signal ratio increases, or equivalently as the problems mentioned above become more acute, the performance of our correlation algorithm degrades gracefully in the following sense:

> As the signal is being degraded, the density of matches decreases accordingly but the ratio of correct to false matches remains high until this proportion has dropped very low.

In other words, a relatively dense disparity map is a *guarantee* that the matches are correct, at least up to the precision allowed by the resolution being used. In this context we also show the effectiveness of a very simple heuristic: if we reject not only invalid matches but also isolated valid matches we can increase even more the ratio correct/incorrect matches without losing a large number of the correct answers. As an example of a possible application of this

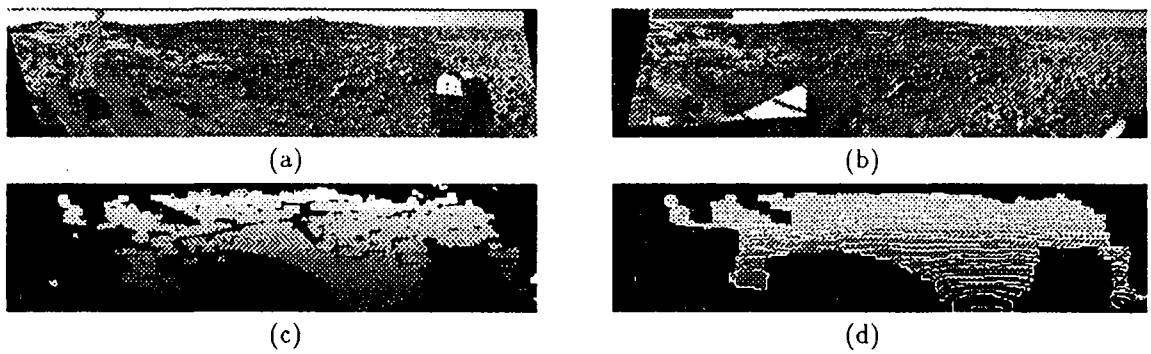


Figure 5: (a)(b) A stereo pair of the martian surface as seen by a Viking lander (c) The disparity map after removal of isolated matches (d) The interpolated depth map with lines of constant w overlaid in white. The black areas are unknown and correspond mostly to areas that are not visible in both images and to the sky. In the depth map, only pixels in areas where the density of valid matches is high are assumed to be known.

desirable feature, in Figure 5, we show a stereo pair of images of the martian surface produced by the viking landers. Note that the part of the ground that can be seen in both images simultaneously is relatively small and that the correlation algorithm naturally produces an almost dense map in this area and an empty one elsewhere. Using this data and the interpolation scheme described in the following section, a mobile robot could compute a very reliable DTM in that area and know that it does not know the shape of the ground in the other areas, which, for safety reasons, would obviously be useful.

Other stereo systems (e.g. [9,12]) include a validity criterion similar to ours but use it as only one among many others. In our case, because we correlate over the whole image and not only at interest or contour points, we do not need the other criteria and can rely on density alone. However, our validity test depends on the fact that it is improbable to make the same mistake twice when correlating in both directions and can potentially be fooled by repetitive patterns (see Figure B.4 in appendix B), which is a problem we have not addressed yet.

2.2 Hierarchical Approach

To increase the density of our potentially sparse disparity map, we use windows of a fixed size to perform the matching at several levels of resolution (computed by subsampling gaussian smoothed images), which is almost equivalent to matching at one level of resolution with windows of different sizes as suggested by Kanade [10] but computationally more efficient. More precisely, as shown by Burt [5], it amounts to performing the correlation using several frequency bands of the image signal.

We then merge the disparity maps by selecting, for every pixel, the (valid) disparity computed at the level of resolution for which the correlation score is highest; in general this will be the finest resolution level for which a disparity that passes our consistency test can be found. In Figure 4 (c) we show the merger of the disparity maps for two levels of resolution that is dense and exhibits more of the fine details of the face than the map of figure 4 (e) computed using only the coarsest level of resolution. The reliability of our validity test allows us to deal very simply with several resolutions without having to introduce, as in [10] for example, a correction factor accounting for the fact that correlation scores for large windows tend to be inferior to those for small windows.

The computation proceeds independently at all levels of resolution and this is a departure from traditional hierarchical implementations that make use of the results generated at low resolution to guide the search at higher resolutions. While this is a good method to reduce computation time, it assumes that the results generated at low resolution are more reliable, even if less precise, than those generated at high resolution; this is a questionable

assumption especially in the presence of occlusions. For example in the case of the trees of Figure 2, it could lead to a computed distance for the area between the trunks that would be approximately the same as that of the trunks themselves, which would be wrong. In appendix A we show that, in the absence of repetitive patterns, the output of our algorithm is not appreciably degraded by using the large disparity ranges that our approach requires.

2.3 Using More Than Two Images

As suggested by several researchers [13,7], more than two images can be used and should be whenever practical. When dealing with three images or more, we take the first one to be our reference frame, compute disparity maps for all pairs formed by this image and one of the others and merge these maps in the same way as those computed at different levels of resolution. In this way we can generate a dense disparity map, such as the one of figure 3 (c): the three images of of Figure 2 belong to a series of five taken by an horizontally moving camera. Taking the image of 2(a) as our reference frame, we merge the four resulting disparity maps, each of them relatively sparse, to produce a dense map with few errors.

In particular, we have been using the INRIA [3] three camera stereo system. To simplify the implementation of our algorithm on a SIMD parallel machine, the images are first reprojected [2] onto the same image plane so that all epipolar lines become parallel. Computing the correlation scores then involves the same sequence of operations at every pixel and becomes easy to implement.

In [2], the authors show that the images can be rectified in such a way as to make the epipolar lines horizontal or vertical, at the expense of a potentially severe deformation. We have found this unnecessary since diagonal epipolar lines can be handled as easily as horizontal ones and we simply take the reprojection plane to be a plane that is parallel to the one passing by the optical centers of the three cameras. Our rectification scheme can be understood as the one that yields parallel epipolar lines with a minimal deformation of the images and is described in detail in appendix B.

2.4 Implementation Issues

The most severe drawback of our approach is its high computational requirement. Our algorithm is implemented in *lisp on a Connection Machine.[tm] [4] As can be seen in Table 2 , for the image sizes we typically deal with such a machine is fast enough to make this problem irrelevant for research purposes. Obviously heuristics, such as a more conventional use of the hierarchy, would have to be used for an implementation on a smaller computer but are not required for any other reason than computational efficiency. Furthermore, correlation is a very regular algorithm that can be implemented in hardware [14] if speed is required at a lower cost. We are currently considering such a hardware implementation of our algorithm and a preliminary study shows that, for the correlation itself, computation times on the order of a second or less are now well within reach. For more details, we refer the interested reader to [6].

In this section we have presented an hypothesis generation mechanism that produces depth maps that are correct where they are dense and unreliable only where they become very sparse. Typically these sparse measurements occur in featureless areas that are usually smooth and at occlusion boundaries where one expects to find an image intensity edge. To compute dense depth maps, one must therefore interpolate those measures in such a way as to propagate the depth information across the featureless areas and preserve depth discontinuities. In the next section, we describe the model and algorithm we use to perform the interpolation.

3 Interpolation

We model the world as made of smooth surfaces separated by depth discontinuities. We also assume that these depth discontinuities produce changes in grey level intensities due to changes in orientation and surface material.

[4] TMC inc.

Window Size	256x256	512x512
3x3	1.75 s (79%)	5.37 s (96%)
5x5	3.14 s (83%)	10.13 s (96%)
7x7	5.03 s (85%)	17.8 s (96%)

Table 2: Computation times required to correlate two images over a range of 50 disparities using a CM2 with a floating point accelerator and 8000 processors. The percentages represent the time actually spent computing on the CM, the remainder being devoted to communicating with the SUN front end. The CM computing time scales linearly with the size of the images and the number of processors while the communication overhead remains approximately constant.

We first describe a simple interpolation model that is well suited to images with sharp contrasts and then propose a refinement of that scheme for lower contrast scenes.

3.1 Simple Interpolation Model

Ideally, if we could measure with absolute reliability the depth, $w0$, at a number of locations in the image, we could compute a depth image w by minimizing the following criterion:

$$C = \int s(w - w0)^2 + \lambda_x(\frac{\partial w}{\partial x})^2 + \lambda_y(\frac{\partial w}{\partial y})^2 \tag{3}$$

$$\begin{aligned} s &= 1 \text{ if } w0 \text{ has been measured}, 0 \text{ otherwise.} \\ \lambda_x &= 0 \text{ if horizontal discontinuity}, c_x \text{ otherwise.} \\ \lambda_y &= 0 \text{ if vertical discontinuity}, c_y \text{ otherwise.} \end{aligned}$$

where c_x and c_y are two real numbers that control the amount of smoothing.

As discussed in the previous section, when a valid disparity can be found it is reliable and can be used, along with the camera models, to estimate $w0$; we then take s to be the normalized correlation score of Equation 1. As shown by Szeliski [17], this amounts to assuming that $w0$ is sampled from the true distance w with a noise whose variance is proportional to $1/s$ i.e.

$$\begin{aligned} w0 &= w + N(0, s^{-1}) \\ \Rightarrow -\log(p(w0|w)) &= 1/2\log(s) + 1/2s(w - w0)\,, \end{aligned} \tag{4}$$

and the $\frac{\partial w}{\partial x}$ and $\frac{\partial w}{\partial x}$ terms come from assuming that the noise is correlated.

Assuming that changes in reflectance can be found at depth discontinuities, we replace the λ_x and λ_y of Equation 3, by terms that vary monotonically with the image gradients in the x and y directions. In fact, we have observed that the absolute magnitudes of the gradients are not as relevant to our analysis as their local relative magnitudes: boundaries can be adequately characterized as the locus of the strongest local gradients, independent of the actual value of these gradients. We therefore write:

$$\begin{aligned} \lambda_x &= c_x Normalize(\frac{\partial I}{\partial x}) \\ \lambda_y &= c_y Normalize(\frac{\partial I}{\partial y}) \end{aligned} \tag{5}$$

where *Normalize* is the piecewise linear function defined by:

$$Normalize(x) = \begin{cases} 1 & \text{if} \quad x < x_0 \\ \frac{x_1 - x}{x_1 - x_0} & \text{if} \quad x_0 < x < x_1 \\ 0 & \text{if} \quad x_1 < x \end{cases} , \tag{6}$$

x_0 and x_1 being two constants. In all our examples, x_0 is the median value of x in the image and x_1 its maximum value . We have also experimented with a *Normalize* function that is proportional to the rank [5] of x and obtained very similar results. The result is also quite insensitive to the value chosen for x_0 as long as it does not become so large as to force the algorithm to ignore all edges. What really matters is the monotonicity of *Normalize* that allows the depth information to propagate faster in the directions of least image gradient and gives to the algorithm a behaviour somewhat similar to that of adaptative diffusion schemes (e.g. [16]).

To compute w we discretize the criterion of Equation 3, yielding

$$\begin{aligned} C &= \sum_{ij} s_{ij}(w_{ij} - w0_{ij})^2 + \lambda_x \sum_{ij}(w_{i+1,j} - w_{i,j})^2 + \lambda_y \sum_{ij}(w_{i,j+1} - w_{i,j})^2 \\ &= S(W - W0)^t(W - W0) + W^t K W \end{aligned} \tag{7}$$

where W and $W0$ are the vectors of all w and $w0$ depths, K the sparse matrix whose "computational molecules" [19] are of the form

$$\begin{vmatrix} 0 & -\lambda_y & 0 \\ -\lambda_x & 2(\lambda_x + \lambda_y) & -\lambda_x \\ 0 & -\lambda_y & 0 \end{vmatrix}$$

and S the diagonal matrix with elements the normalized correlation scores at every point (0 where the matches are invalid). We then use a conjugate gradient method [18,19] to solve the equation

$$\begin{aligned} \frac{\partial C}{\partial W} &= 0 \\ \Rightarrow (K + S)W &= SW0 \quad . \end{aligned} \tag{8}$$

The *lisp implementation of the conjugate gradient method involves only NEWS nearest neighbor communication and, here again, it is possible to develop [11] specialized hardware if speed is required.

In Figure 6, we show the depth map computed by interpolating the disparity map of Figure 4(c) and three views generated by illuminating this map from different directions. Note that the main features of the face, nose, eyebrows and mouth have been correctly recovered.

In Figure 7, we show the behaviour of our algorithm on a synthetic image with a central square that presents a ramp in intensity and is corrupted by a gaussian noise. The central square is shifted by a constant disparity in a second image resulting, after correlation, in the disparity map of 7(b) where the black pixels are those for which no valid match can be found (mainly the pixels that are occluded in the second image). In Figure 7(c) the rounded curve is a plot of the interpolated depths along a horizontal line passing through the center of the image. The depth discontinuities are well preserved where the contrast is sharp but tend to be slightly blurred where the contrast becomes low. This interpolation technique is therefore appropriate for the face of Figure 4 that presents few low-contrast depth discontinuities but produces a somewhat blurry result for the tree scene of Figure 2, as can be seen in Figure 8(a). To improve upon this situation, we propose a slightly more elaborate interpolation scheme that takes depth discontinuities explicitly into account.

3.2 Introducing Depth Discontinuities

The λ_x and λ_y coefficients defined by Equation 5 introduce "soft" discontinuities: when the contrast is low, some smoothing occurs across the discontinuity. The depth image, however, is less smoothed than in the complete absence of an edge resulting in s strong w gradient at such depth discontinuities. We take advantage of this property of our "adaptative" smoothing by defining the following iterative scheme:

[5] Computed by ordering the values of x in the image and assigning to x a value between 0 and 1 that is proportional to its rank.

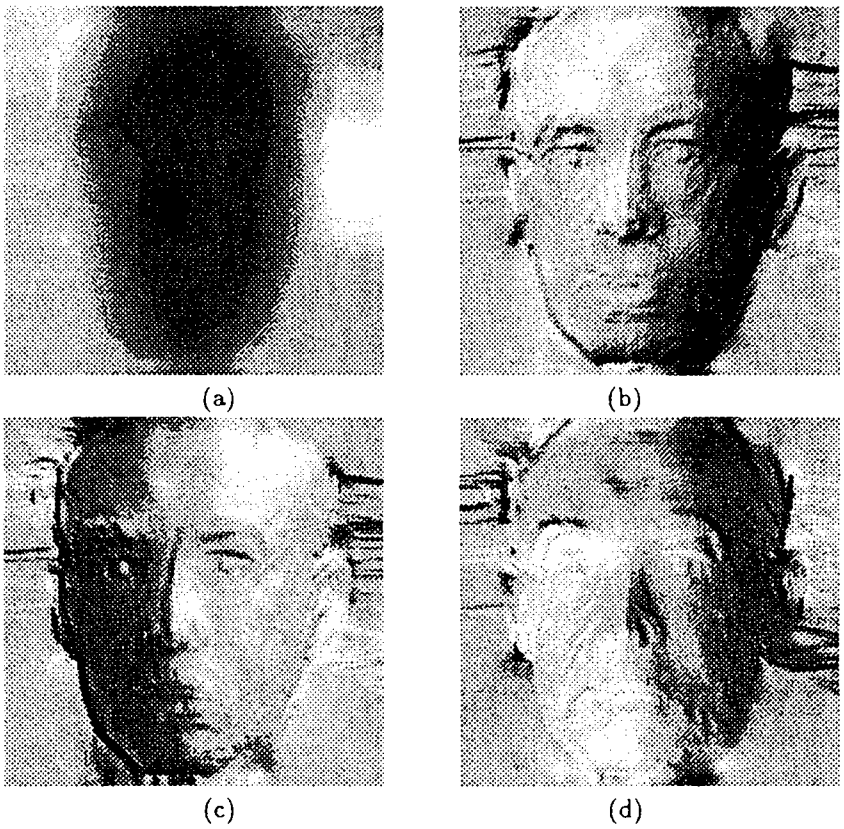


Figure 6: (a) Interpolated depth map for the face of Figure 4 (b) (c) and (d) Shaded views generated by shining a light from three different directions.

1. Interpolate using the λ_x and λ_y defined above.
2. Iterate the following procedure:
   (a) Recompute λ_x and λ_y as functions of both the intensity gradient and the depth gradient of the interpolated image:

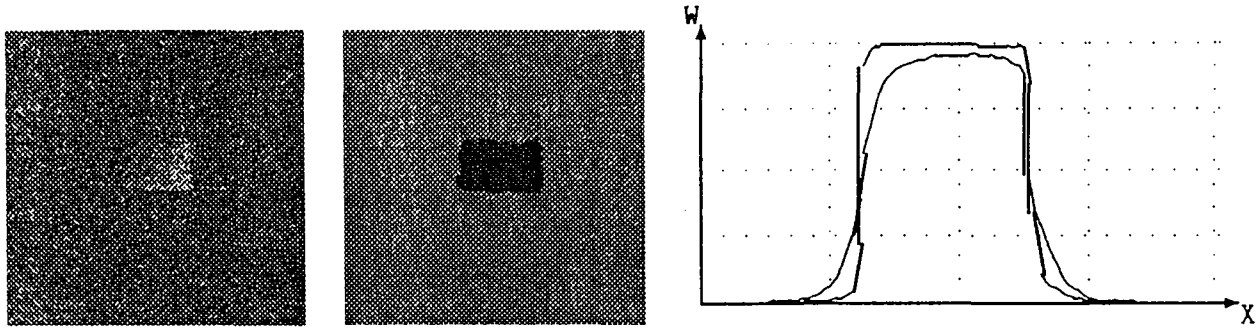
$$\begin{aligned} \lambda_x &= Normalize(\frac{\partial I}{\partial x})Normalize(\frac{\partial w}{\partial x})^\alpha \\ \lambda_y &= Normalize(\frac{\partial I}{\partial y})Normalize(\frac{\partial w}{\partial y})^\alpha \end{aligned} \tag{9}$$

   where α is a constant equal to 2 in our examples.

   (b) Interpolate again the raw disparity map using the new λ_x and λ_y coefficients.

The algorithm converges in a small number of iterations resulting in a much sharper depth map. The squarish curve of Figure 7(c) is a plot of the depth interpolated from the disparity map of Figure 7(b) after four iterations. Similarly in Figure 8(b), we show a much improved depth map for the tree scene after the same number of iterations.

In Figure 9, we show another example of this behaviour on an indoor scene that is a difficult one for a correlation based algorithm because there is little texture outside of the label of the bottle. As discussed in section 2, the



(a) (b) (c)

Figure 7: (a) Synthetic image with a central square that presents a ramp in intensity and gaussian noise. (b) The disparity map computed by correlation (c) A slice through a portion of the smoothed image.

disparity map is dense where the label is and sparse elsewhere resulting in the blurry depth map of Figure 9(c) after interpolation using coefficients defined in the previous subsection and the map of map Figure 9(d) where the bottle clearly stands out after four iterations of the iterative smoothing scheme.

4 Conclusion

In this work we have described a correlation based algorithm that combines two simple and parallel techniques to yield reliable depth maps in the presence of depth discontinuities, occlusions and featureless areas:

- The correlation is performed twice over the two images by reversing their roles and only matches that are consistent in both directions are retained, thereby guaranteeing a very low error rate.
- The disparity map is then interpolated using a technique that takes advantage of the grey level information present in the image to preserves depth discontinuities and propagate the information across featureless areas.

The depth maps that we compute are qualitatively correct and the density of acceptable matches provides us with an excellent estimate of their reliability. Because of the great regularity and simplicity of the techniques described here, we hope to be able to build dedicated hardware that would implement them and could, for example, be used by a mobile robot in an outdoor environment.

Furthermore because the reliability of the depth maps is easy to assess, a system based on our algorithm would know when to invoke additional sources of three dimensional information, such as geometrical constraints, shape from shading or the output of an active ranging sensor, to fill in those areas of uncertainty. In future research we intend to investigate the possibilities that such an approach has to offer.

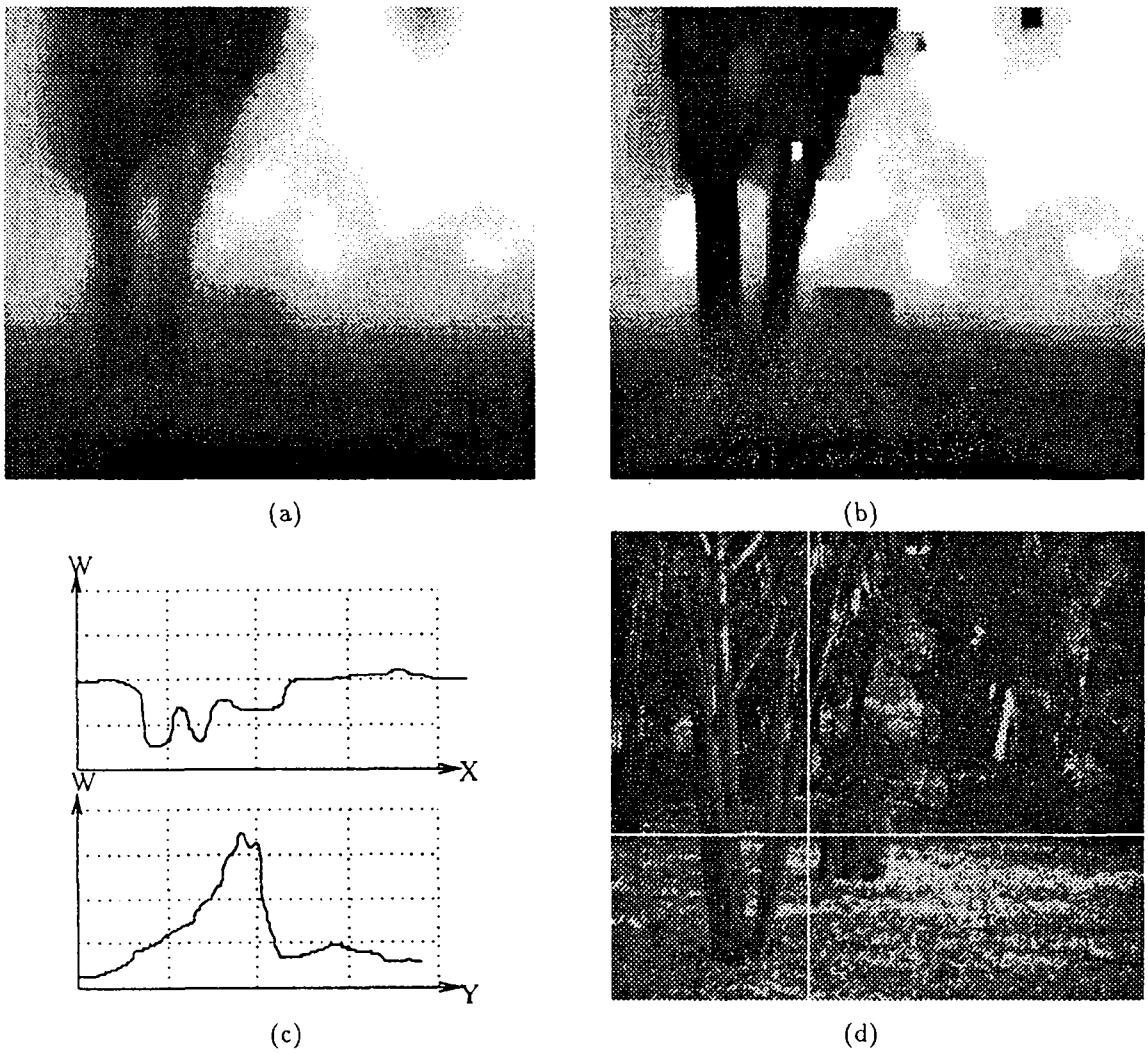


Figure 8: (a) Trees depth image computed by smoothing using the algorithm of section 3.1. (b) Depth image after four iterations of the iterative scheme of section 3.2. (c) Depth values along the horizontal and vertical lines plotted in (d). We have stretched the depth images to enhance the contrast so that the furthest areas appear completely white. Note that the trunks and the stump clearly stand out.

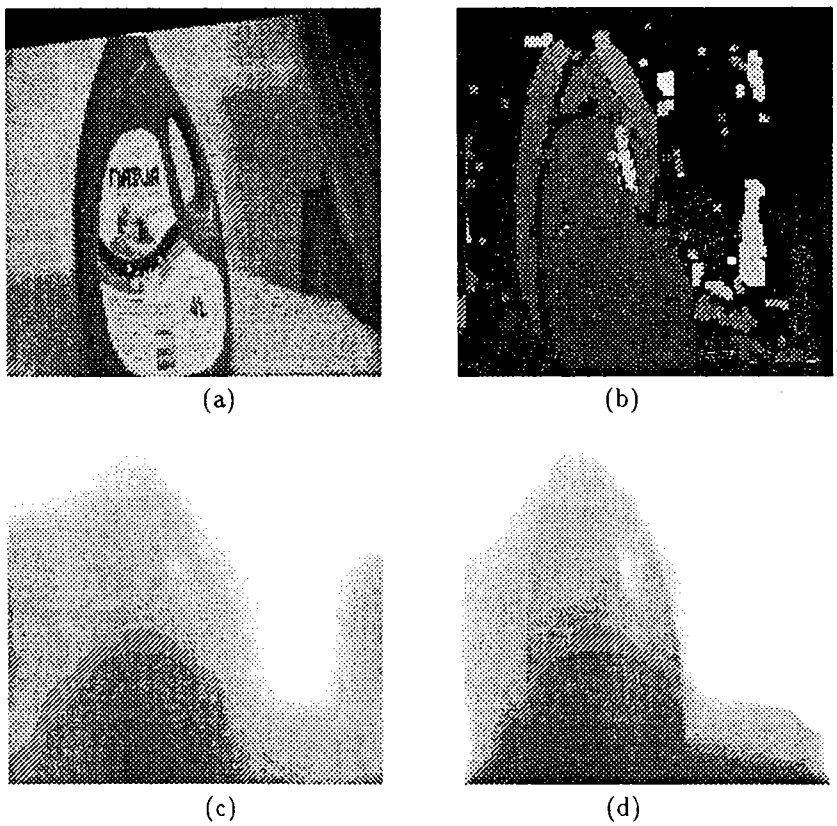


Figure 9: (a) One image taken from an triplet, note that only the label of the bottle is textured. (b) The corresponding disparity map. (c) The interpolated depth map of computed using the coefficients defined in section 3.1. (d) The interpolated depth map after four iterations of the interpolation scheme of section 3.2. The depth images have been stretched as in Figure 8.

Appendices

A Behaviour of the Correlation Algorithm on Synthetic Data

In this appendix we model the behaviour of our correlation algorithm using synthetic data and show that the validity test defined in section 2 allows our algorithm to make few mistakes and forces it to produce very sparse maps when the data becomes too noisy and the matches unreliable.

In the remainder of this section, we use a stereo pair formed by two synthetic images I_1 and I_2 defined as follows:

$$\begin{aligned} I_1 &= N_0(0, \sigma_{texture}) + N_1(0, \sigma_{noise}) \\ I_2 &= N_0(0, \sigma_{texture}) + N_2(0, \sigma_{noise}) \end{aligned} \tag{A.1}$$

where N_0, N_1 and N_2 are three independent gaussian random variables of variance $\sigma_{texture}$ and σ_{noise} and we define the noise to signal ratio

$$n/s = \sigma_{noise}/\sigma_{texture} \tag{A.2}$$

such that the two images are identical when n/s is zero and that the correlation is degraded as n/s grows. To gauge the performance of our correlation algorithm we introduce two functions, f_{valid} the proportion of pixels for which a valid match (according to our criterion) can be found, and f_{error} the proportion of pixels among these for which the match is erroneous, that is for which the computed disparity [6]is different from zero. These two functions depend only on

- the noise to signal ratio,
- the size of the correlation windows,
- the range of disparities being tested.

Below we show the influence of these parameters using curves that have been computed by running large simulations on the Connection Machine.^{tm}

A.1 Influence of the noise to signal ratio

In figure A.1 (a) we plot f_{valid} as a function of n/s for three different window sizes and for a fixed disparity range of twenty integer disparities centered around 0. Similarly, in A.1 (b), we plot f_{error}. f_{error} increases with n/s while f_{valid} decreases towards the probability of a match in the absence of a signal, which is very low for the 7x7 and 5x5 windows. For these window sizes, f_{error} does not become significant before f_{valid} has dropped below about 25% justifying our claim that the density of the disparity map can be regarded as a confidence estimate. The general behaviour of the two functions for 3x3 windows is fundamentally the same but the probability of a match in the absence of a signal is now non negligeable and only very dense disparity maps can be regarded as reliable if they have been computed with such small windows.

For comparison's sake, in Figure A.2 we show the probability of error when the correlation is performed only from I_1 to I_2 without imposing our validity criterion. Note that the proportion of errors becomes significant much earlier. In short, at the cost of loosing a small number of the correct matches, our technique allows us dispose almost completely of the erroneous ones, at least for sufficiently large correlation windows.

If we are willing to accept somewhat sparser disparity maps, we can increase the reliability of the correlation algorithm even more by removing the isolated and probably erroneous matches. To do so we treat the disparity map as a binary array in which valid matches are represented as ones and invalid ones as zeros that we then shrink and reexpand it to remove isolated points. In Figure A.1(c) and (d) we plot f_{valid} and f_{error} after having shrunk and reexpanded the maps by one pixel. For the larger windows, the ratio of errors does not become significant until f_{valid} has dropped almost all the way to zero indicating that the removed points were almost all in error.

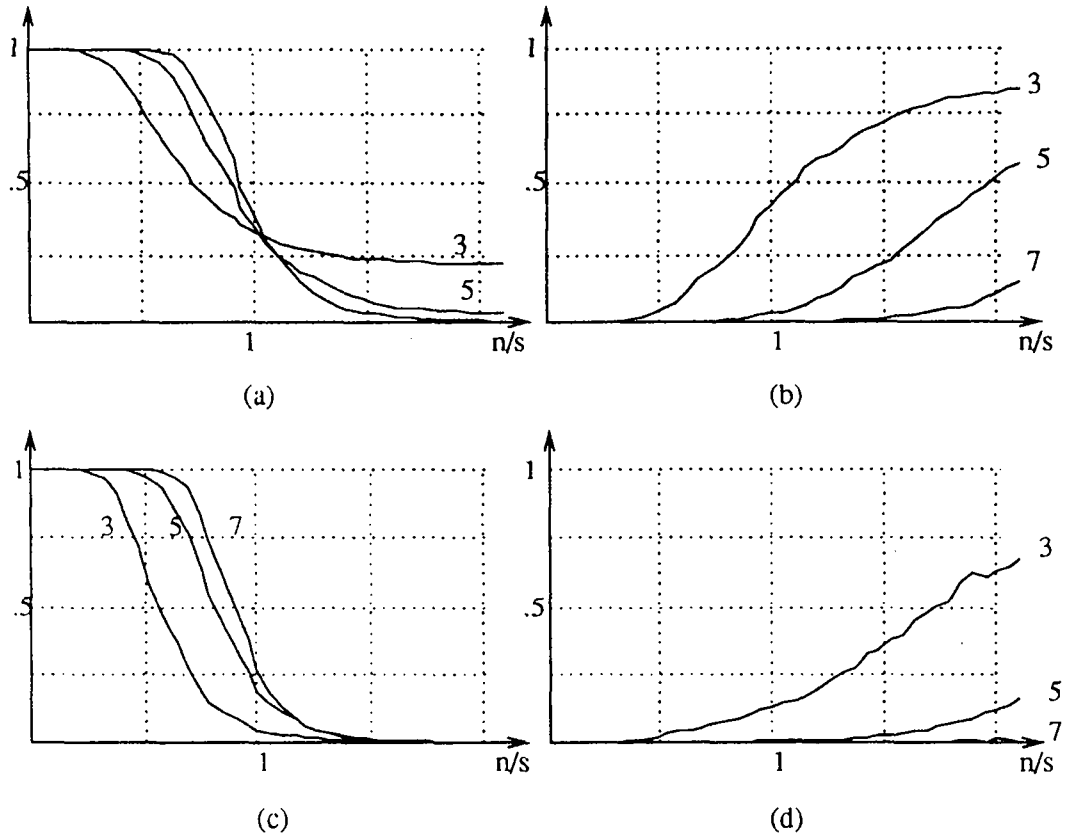


Figure A.1: (a) The proportion of matched pixels for three window sizes, 3x3, 5x5 and 7x7, as a function of the noise to signal ratio. (b) The proportion of incorrectly matched pixels as a function of the noise to signal ratio. (c) and (d) The proportions of matched pixels and incorrect matches after removing isolated matches.

A.2 Influence of the size of the disparity range

In Figure A.3 (a) (b) we plot f_{valid} and f_{error} computed using three sizes, 10, 20 and 40, of the disparity range and 5x5 windows. In Figure A.3 (c) (d) we plot the same curves for 3x3 windows. For low values of the noise to signal ratio the proportion of matched pixels is slightly less for large disparity intervals because more good matches are "lost" accidentally. For high values of n/s the proportion of errors increases somewhat for large disparity ranges because the chances of an accidental match also increase. Thus, the performance of the algorithm is somewhat degraded when the disparity range increases but, all in all, the effect is quite minor and almost insignificant for large windows. This is why we can get good results with our simple hierarchical scheme that does not use the results found at the coarsest resolutions to guide the search at the finest ones.

[6] For the purpose of this test we use integer disparities and do not interpolate.

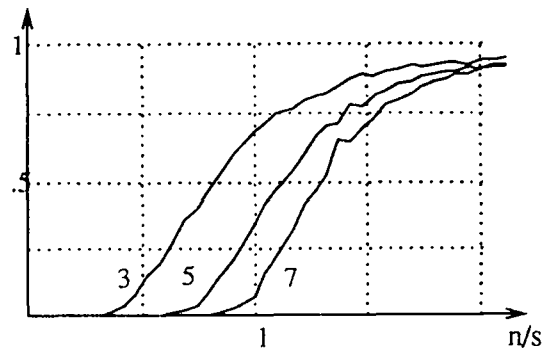


Figure A.2: The probabilities of error for the three window sizes as a function of the noise to signal ratio when no validity test is performed

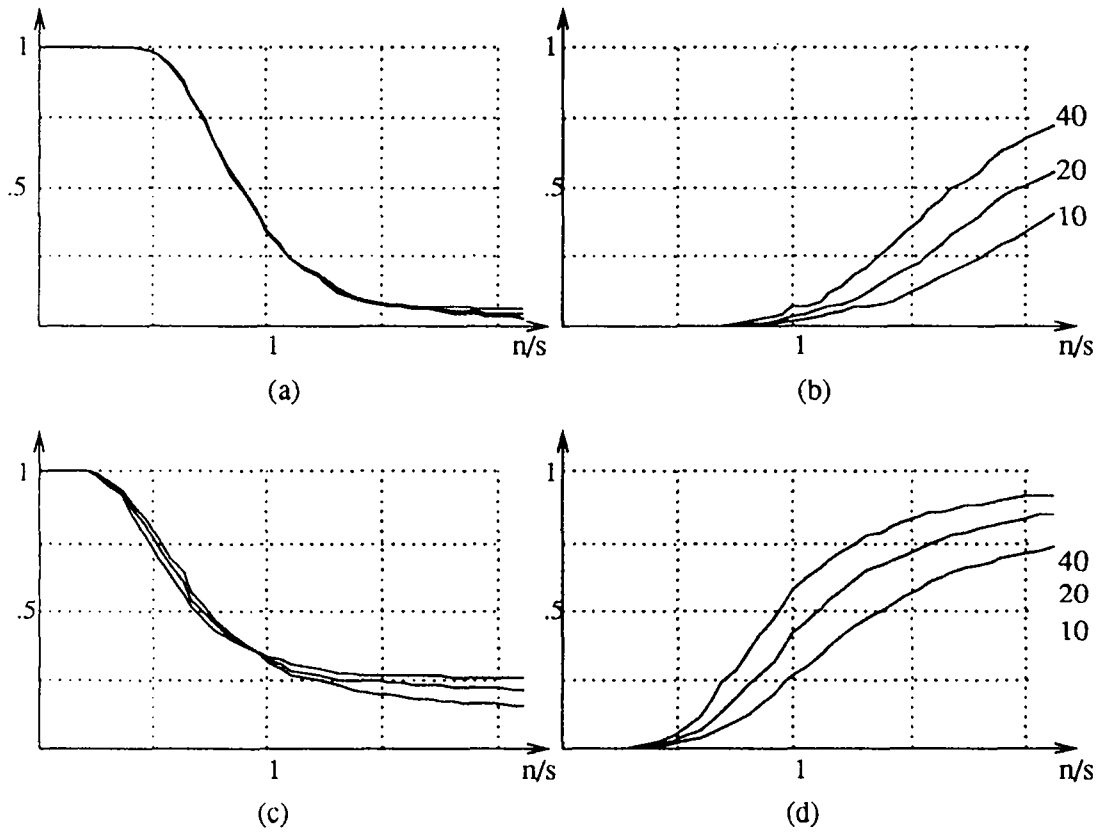


Figure A.3: The proportions of matched pixels and incorrect matches as a function of the signal to noise ration for three sizes of the disparity range: 3x3 windows (a) and (b), 5x5 windows (c) and(d)

B Rectification

In this appendix, we describe the rectification techniques we use to deal with triplet of images produced by the INRIA 3 camera stereo system. For a more thorough description of the mathematical formalism used here, we refer the interested reader to the article by Ayache et al.[2].

B.1 From Image Planes to Calibration Matrices

Each camera is modeled, using the classic pinhole model, by its optical center C, its image plane $\mathcal{P}$ and a 4x3 calibration matrix T such that if the image point $I = (u, v)$ is the projection of the world point $P = (x, y, z)$ the following relationships hold:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = T \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \qquad (B.1)$$

$$u = U/W$$
$$v = V/W \quad .$$

T is such that

$$TC = 0 \quad . \qquad (B.2)$$

and given the center $C = (x_c, y_c, z_c)$, the plane $\mathcal{P}$, its origin and axes, T can be derived as follows. Let

$$ax + by + zc = h \text{ where } a^2 + b^2 + c^2 = 1 \qquad (B.3)$$

be the equation of plane P and let M_0 and T_0 be two 4x4 matrices:

$$M_0 = \begin{pmatrix} h & 0 & 0 & a \\ 0 & h & 0 & b \\ 0 & 0 & h & b \\ 0 & 0 & 0 & 0 \end{pmatrix} , \qquad (B.4)$$

$$T_0 = \begin{pmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

M_0 is such that, given a world point P with projective coordinates $(x, y, z, 1)$, the point MP is the intersection of the plane $\mathcal{P}$ and the line going through P and the world origin 0. T_0 is the matrix representing the translation of vector $\overrightarrow{OC}$. Using the pinhole camera model, it is easy to see that the matrix

$$M = T_0 M_0 T_0^{-1} \qquad (B.5)$$

is such that for a world point P, $I = MP$ is the image point that is the projection of P through the camera optical center as shown in Figure B.1. The projective image coordinates of I can be computed from its world coordinates by multiplying them by a 3x4 matrix, N, that depends only on the arbitrarily chosen axes and origin of the plane P but not on the camera. The calibration matrix T is then taken to be

$$T = NM \qquad (B.6)$$

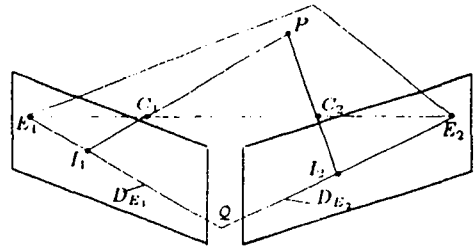


Figure B.1: Pinhole model for two cameras: C_1 and C_2 are the optical centers of the two cameras and the world point P projects to I_1 and I_2 respectively. E_1 and E_2 are the epipoles through which all epipolar lines go and D_{E_1} and D_{E_2} are the epipolar lines on which I_1 and I_2 are bound to lie.

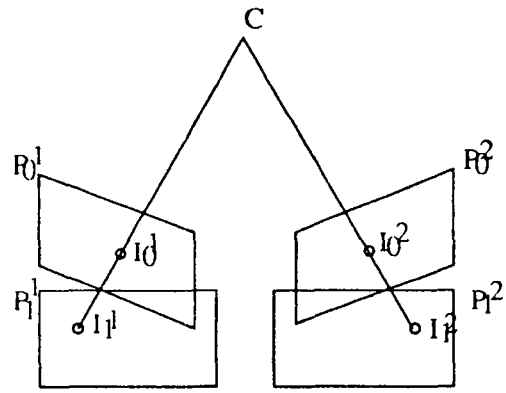


Figure B.2: Rectification of two images: the two original images I_0^1 and I_0^2 are rectified into I_1^1 and I_1^2 by reprojecting them to the same images plane $P_1^1 = P_1^2$.

B.2 Rectification Matrices

Given several images and a point in one of them, the corresponding points in the other images are bound to lie on epipolar lines and these epipolar lines are parallel if and only if all the image planes are parallel. In our application, we rectify the three images by reprojecting the them from their respective image plane $\mathcal{P}_0$ to a plane $\mathcal{P}_1$ that is parallel to the one passing by the three optical centers without changing the optical center as shown in Figure B.2. To every image point of the original image plane corresponds a unique point of the rectified one and we derive below their analytical relationship.

Let $T_0 = [R_0, C_0]$ and $T_1 = [R_1, C_1]$ be the two corresponding 3x4 calibration matrices computed as described above, where R_0 and R_1 are 3x3 matrices and C_0 and C_1 3x1 matrices. Let $I_0 = (U_0, V_0, 1)$ be a point of the original image and $I_1 = (U_1, V_1, W_1)$ the corresponding one in the rectified image. Let then P be a world point that projects at both I_0 and I_1, i.e.

$$\begin{aligned} I_0 &= T_0 P \\ I_1 &= T_1 P \end{aligned} \tag{B.7}$$

We write P as $C + \lambda(x, y, z, 1)$, where C is the common optical center and λ a real number. Because

$$T_0 C = T_1 C = 0 \tag{B.8}$$

we can write

$$\begin{aligned}
\begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} &= \lambda T_0 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \lambda(R_0 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + C_0) \\
\begin{pmatrix} U_1 \\ V_1 \\ W_1 \end{pmatrix} &= \lambda T_1 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \lambda(R_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + C_1) \\
\Rightarrow \begin{pmatrix} U_1 \\ V_1 \\ W_1 \end{pmatrix} &= R_1 R_0^{-1}(\begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} - C_0) + wC_1 \\
&= R_1 R_0^{-1} \begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} + (C_1 - R_1 R_0^{-1} C_0) \\
&= Rect \begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix}
\end{aligned} \tag{B.9}$$

where *Rect* is the 3x3 matrix computed by adding to the last column of $R_1 R_0^{-1}$ the vector $C_1 - R_1 R_0^{-1} C_0$.

B.3 Rectifying the images

Having computed the rectification matrix *Rect* of equation B.9 and its inverse, we can now transform the images. Given a point I_1 of the rectified image, its corresponding point in the original image is $I_0 = Rect^{-1} I_1$ in the original image and we compute the grey level of I_1 using bilinear interpolation. In Figure B.3 we show a triplet of images and the corresponding images after rectification. The grid that appears in all three views is used to compute the original calibration matrices [20]. Note that the rectified images are not very deformed because the three original image planes were not too far from being parallel. In Figure B.4, we show the output of our stereo algorithm.

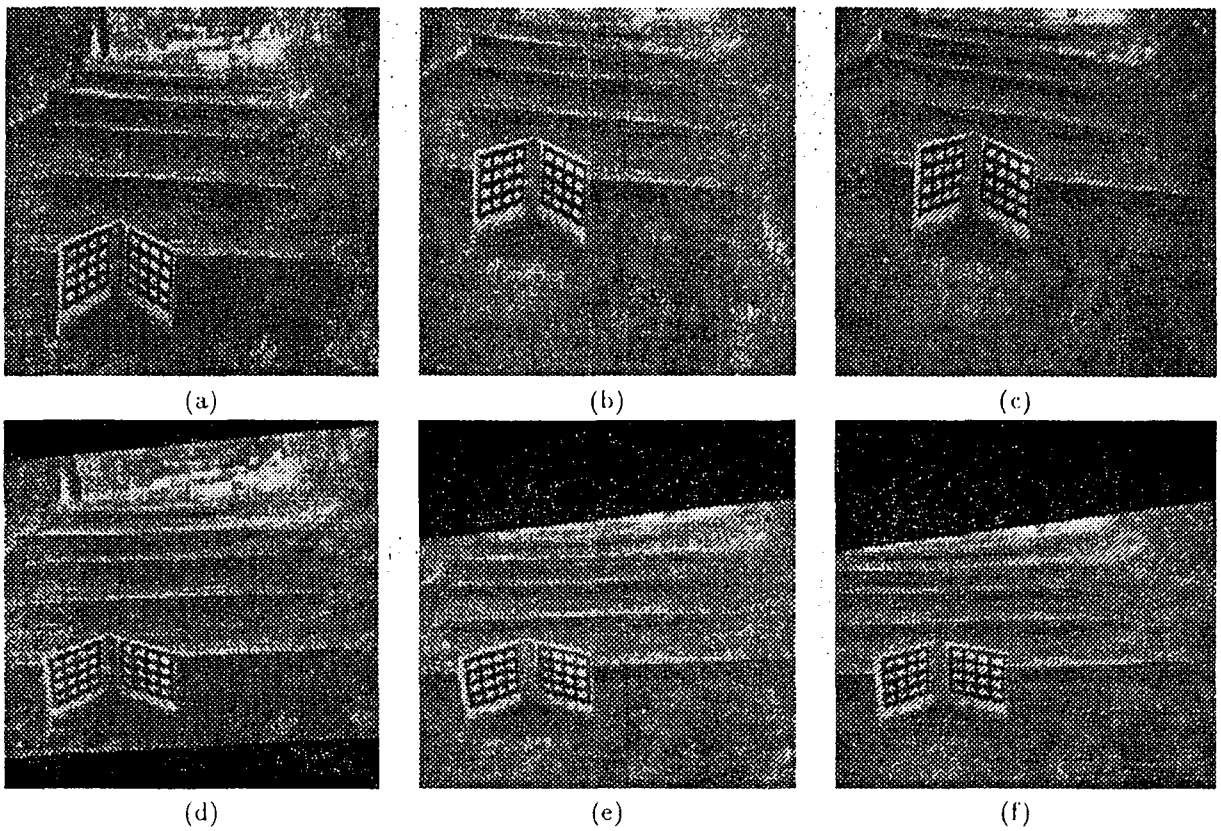


Figure B.3: (a) (b) (c) A triplet of images. (d) (e) (f) The images after rectification. The grid in the bottom left corner is used to calibrate the system ad compute the T matrices.

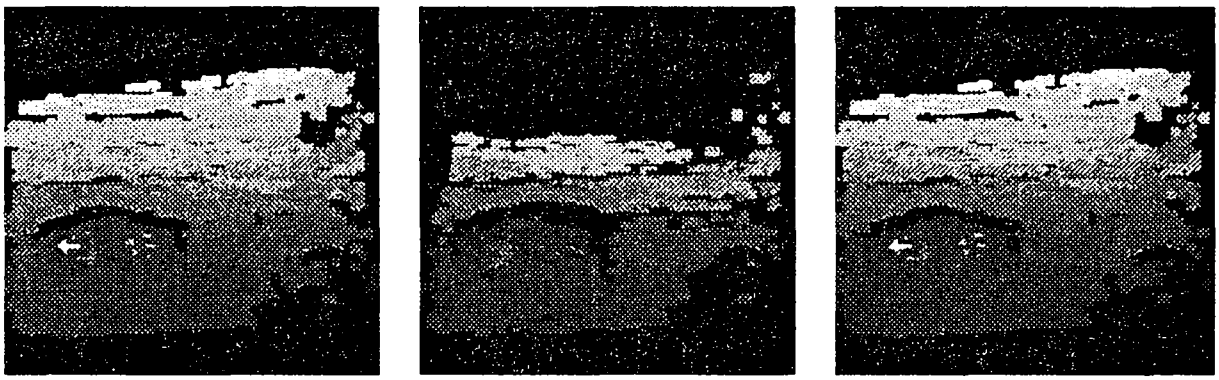


Figure B.4: Disparity maps: (a) computed by correlating B.3 (a) with B.3 (b), (b) by correlating B.3 (a) with B.3 (c) and (c) merging the two maps. These maps are virtually error free except for those caused by the repetitive patterns on the grid itself.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[2] N. Ayache. and C. Hansen. Rectification of images for binocular and trinocular stereovision. In *Ninth International Conference on Pattern Recognition*, pages 11–16, Rome, Italy, November 1988.

[3] N. Ayache and F. Lustman. Fast and Reliable Passive Trinocular Stereovision. In *First International Conference on Computer Vision*, June 1987.

[4] S.T. Barnard and M.A Fischler. Computational stereo. *Computational Surveys*, 14(4):553–572, 1982.

[5] P.J. Burt, C. Yen, and X. Xu. Local correlation measures for motion analysis. In *IEEE PRIP Conference*, pages 269–274, 1982.

[6] C. Cailler, F-X. Fornari, P. Heng, and T. Holtzer. *Cocosun*. Rapport de Stage, Cerics, December 1990.

[7] O.D. Faugeras. *A Few Steps Toward Artificial 3D Vision*. Rapport de Recherche 790, INRIA, February 1988.

[8] E. Güelch. Results of test on image matching of isprs wg iii / 4. *International Archives of Photogrammetry and remote sensing*, 27(III):254–271, 1988.

[9] M.J. Hannah. Digital stereo image matching techniques. *International Archives of Photogrammetry and remote sensing*, 27(III):280–293, 1988.

[10] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptative window: theory and experiment. In *Image Understanding Workshop*, September 1990. Available as Technical Report CMU-CS-90-120 from CMU computer science department.

[11] C. Mead. Analog vlsi for auditory and vision signal processing. In *INSPEC Conference*, San Francisco, California, December 1988.

[12] A. Meygret, M. Thonnat, and M. Berthod. A pyramidal stereovision algorithm based on contour chain points. In *ECCV90 Conference*, Antibes, France, April 1990.

[13] H. Moravec. *Robot Rover Visual Navigation*. UMI Research Press, Ann Arbor, Michigan, 1981.

[14] H.K. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5), 1984.

[15] H.K. Nishihara and T.Poggio. Stereo vision for robotics. In *ISRR83 Conference*, Bretton Woods, New Hampshire, 1983.

[16] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *IEEE Computer Society Workshop on Computer Vision*, pages 16–22, Miami, Florida, 1987.

[17] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Press, Norwell Massachusetts, 1989.

[18] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, June 1990.

[19] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139, March 1986.

[20] G. Toscani, R. Vaillant, R. Deriche, and O.D. Faugeras. Stereo Camera Calibration Using The Environment. In *6th Scandinavian Conference on Image Analysis*, pages 953–960, 1989.

Imprimé en France
par
.l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249 - 6399

ISSN 0249-6399