Machine Vision and Applications (1993) 6:35–49

Machine Vision and Applications

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

Pascal Fua[1,2]

[1] INRIA Sophia-Antipolis, 2004 Route des Lucioles, F-0656 Valbonne Cedex, France
[2] SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA

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 eliminate gross errors: at places where the images cannot be correlated reliably, due to lack of texture of 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 a 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.

Key words: Correlation – Optimization – Parallel – Regularization – Stereo

1 Introduction

Over the years numerous algorithms for passive stereo have been proposed. They can roughly be classified in two categories (Barnard and Fischler 1982):

1. Feature-based. These 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 only yield very sparse depth maps.
2. Area-based. 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. The resulting depth map can then be interpolated. The underlying assumption appears to be a valid one for relatively textured areas; however, it may prove wrong at occlusion boundaries and within featureless regions.
Alternatively, the map can be computed by directly fitting a smooth surface that accounts for the disparities between the two images. This is a more principled approach since the problem can be phrased as one of optimization; however, the smoothness assumptions that are required may not always be satisfied.

All these techniques have their strengths and weaknesses and it is difficult to compare their 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 (1988). In this work, the author has assembled a standardized data set and sent it to 15 research institutes across the world. It appears that the correlation-based system developed at SRI by Hannah (1988) has produced the best results both in terms of precision and reliability. This system achieves precisions in the order of half a pixel in disparity, but, unfortunately, 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 fery 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

* This research was supported in part under the Centre National d'Etudes Spatiales VAP contract and in part under a Defence Advanced Research Projects Agency contract at SRI

and fit a surface that is piecewise smooth. These algorithms have proven very effective on real data. Their parallel implementation on a Connection Machine[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 all image points and retain only matches 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 similarity 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 scores is generated for integer disparity values. We use a measure based on normalized mean-squared differences of grey level values. In the remainder of the paper, following Brown and Ballard (1982), we will refer to this score s as correlation score and take it to be:

$$s = \max(0, 1 - c) \tag{1}$$

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

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 could then be taken to be the one that provides the highest value of s[2]. In fact, to compute the disparity with subpixel accuracy, we fit a second degree curve to the correlation scores in the neighborhood of the optimum and compute the optimal disparity by interpolation.

For comparison's sake, we have also implemented a normalized cross correlation measure that occasionally yields slightly different results. However, after applying the smoothing algorithm of Sect. 3, the depth images computed using the mean-squared differences method and normalized cross correlation become undinstinguishable.

Thanks to the subpixel interpolation, both methods yield disparities with precisions of better than a pixel on average. Their respective behaviors are discussed briefly in Appendix B.

[1] Trademark: TMC Inc.

[2] Imposing $s > 0$ amounts to choosing a very weak threshold on the correlation measure

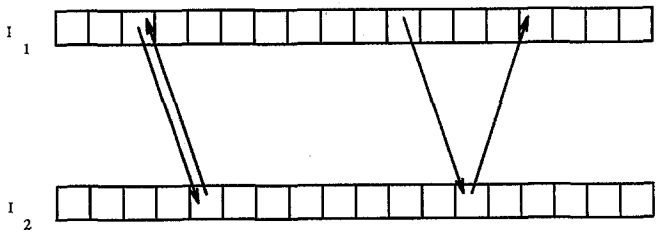


Fig. 1. Consistent vs inconsistent matches: the two *rows* represent pixels along two epipolar lines of I_1 and I_2 and the *arrows* go from a point in one of the images towards the point in the other image that maximizes the correlation score. The match on the *left* is consistent because correlating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

2.1 Validity of the disparity measure

As shown by Nishihara and Poggio (1983), 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 scene 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 (Anandan 1989). It is more robust, but also involves a 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 use small windows reliably. 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 Fig. 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 scores when correlating the window centered on P_2 with windows that shift along the epipolar line of I_1 corresponding to P_2.

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 behavior using the portion of the tree scene of Fig. 2 outlined in Fig. 2a. Different parts of the ground between the two trees and between the trees and the stump are occluded in Fig. 2b and c. In Fig. 3a and b, we show the computed disparities for this image window after correlation with the images shown in Fig. 2b and c, respec-

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

Fig. 3. a The result of matching 2a and 2b for window of 2a delimited by the white rectangle. **b** The result of matching 2a and 2c for the same windows. **c** The merger of four disparity maps computed using the image of Fig. 2a as a reference frame, the two other images of Fig. 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 there are few false matches although the correlation windows used in this case are very small (3×3)

tively. 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 3×3 correlation windows; these small windows are sufficient in this case because the scene is highly 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 it is a distinct advantage to be able to use small windows, because the correct depth of the gound behind the trees could not be computed with larger ones that would include the tree trunks.

We use the face shown in Fig. 4 to demonstrate another case in which the validity test rejects false matches. The epipolar lines are horizontal and in Fig. 4d we show the resulting disparity image, using 7×7 windows, in which the invalid matches appear in black. In Fig. 4e we show another disparity image computed after one of the images has been shifted 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 behavior 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 algorithm tends, instead of making mistakes, to yield sparse maps. This actually is a very generic behavior 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. These algorithms are known to fail when:

- 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

[3] Assumed not to be exactly vertical

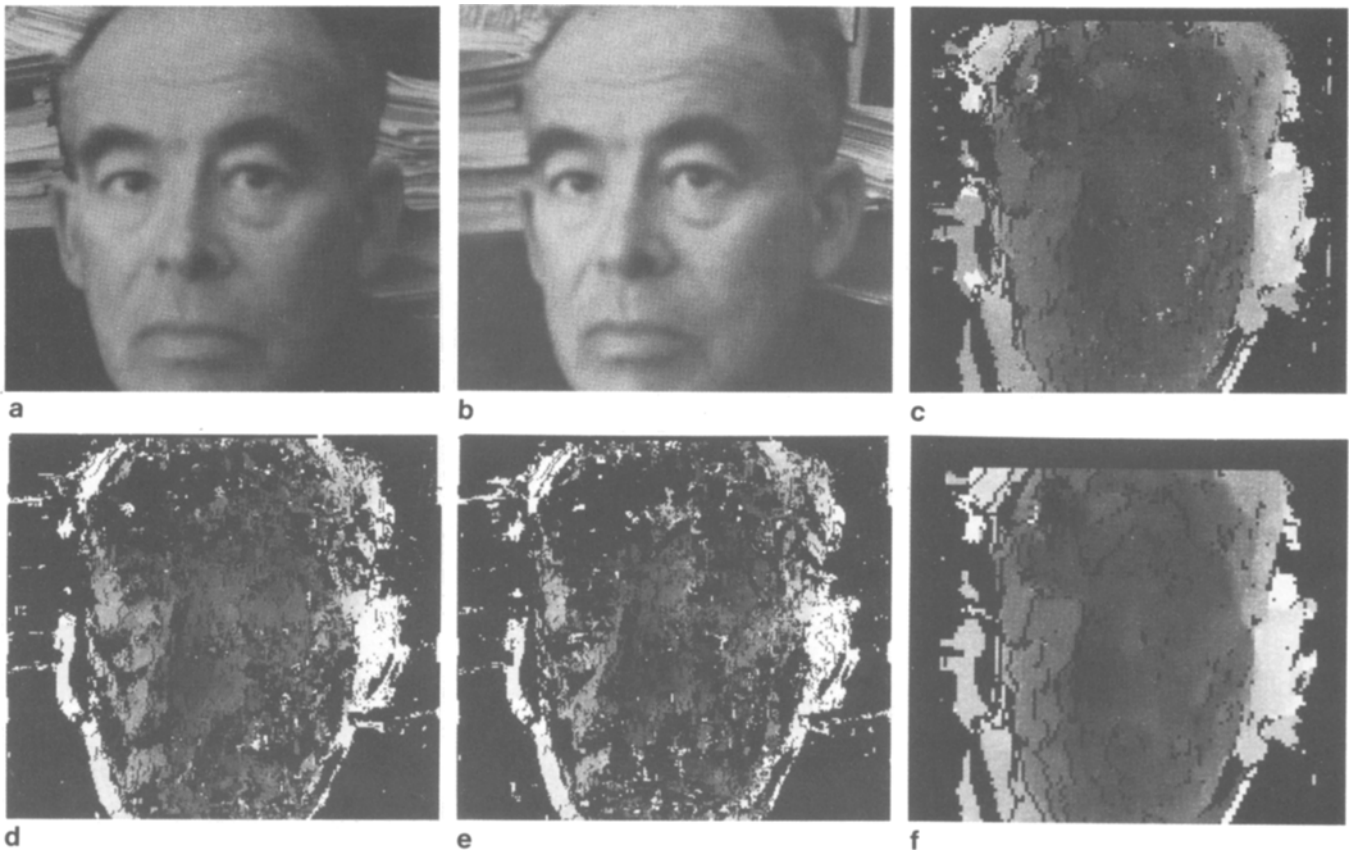


Fig. 4. **a** Left and **b** right 256 × 256 images of a face. **c** The disparity map obtained by merging the results computed at two levels of resolution. **d** Disparity map computed at the highest resolution. **e** Disparity map computed at the highest resolution after shifting the right image up by two pixels. **f** Disparity map computed at the lower resolution

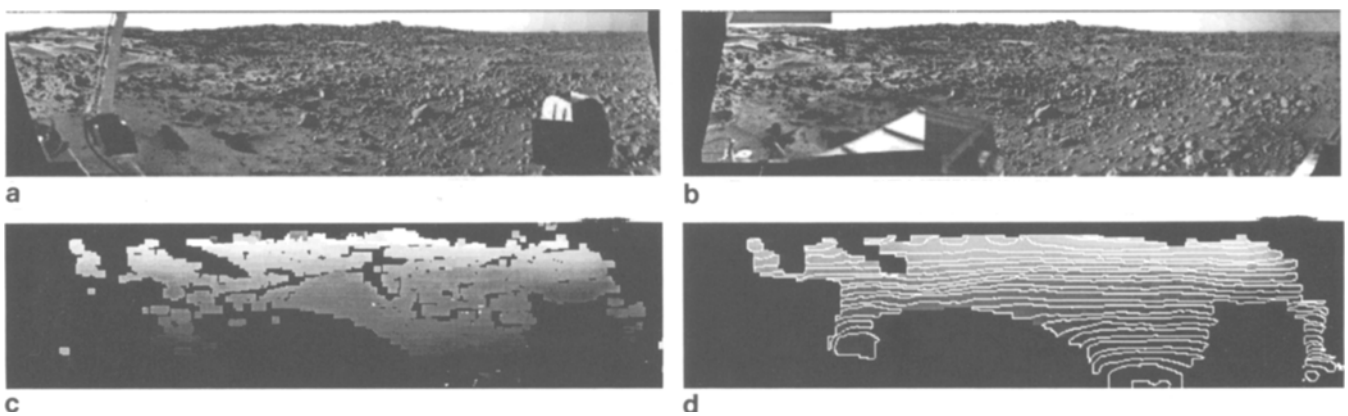


Fig. 5. **a,b** Stereo pair of the Martian surface as seen by a Viking lander. **c** The disparity map after removal of isolated matches. **d** The interpolated depth map with lines of constant w overlaid in white. The *black areas* are unknown and correspond mostly to areas that are not visible in both 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

that corrupts the signal. In Appendix A, we use synthetic data to show that as the signal-to-noise ratio decreases, 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 the disparity map becomes very sparse. 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 of correct/incorrect matches without losing a large number of the correct answers. As an example of a possible application of this desirable feature, in Fig. 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. Hannah 1988; Meygret et al. 1990) 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. Thus, it can potentially be fooled by repetitive patterns (see Fig. C4 in Appendix C), 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,[4] which is almost equivalent to matching at one level of resolution with windows of different sizes (Kanade and Okutom, 1990) but computationally more efficient. More precisely, as shown by Burt et al. (1982), it amounts to performing the correlation using several frequency bands of the image signal.

As discussed in Appendix B, the best precisions are obtained for the smaller windows or, equivalently, the higher levels of resolution. We therefore merge the disparity maps by selecting, for every pixel, the highest level of resolution for which a valid disparity has been found. In Fig. 4c we show the merger of the disparity maps for two levels of resolution. This merger is dense and exhibits more of the fine details of the face than the map of Fig. 4e 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 Kanade and Okutomi (1990) 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 these are good methods for reducing computation time, they assume 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 Fig. 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 range that our approach requires.

2.3 Using more than two images

As suggested by many researchers, including Faugeras (1988) and Moravec (1981), more than two images can and should be used 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 other images, and then 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 Fig. 3c: the three images of Fig. 2 belong to a series of five taken by a horizontally moving camera. Taking the image of 2a 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 Ayache and Lustman (1987) three-camera stereo system. To simplify the implementation of our algorithm on a SIMD parallel machine, the images are first reprojected (Ayache and Hansen 1988) 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.

Ayache and Hansen (1988) 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. 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. This approach is described in detail in Appendix C.

2.4 Implementation issues

The most severe drawback of our approach is its high computational requirement. Our algorithm is implemented on a Connection Machine.[5] For the image sizes we typically deal with, such a machine is fast enough to make this problem irrelevant for research purposes (Table 1). Our implementation on a SPARCstation 2[6] workstation runs in approximately

[4] Computed by subsampling gaussian smoothed images

[5] Trademark: TMC Inc.

[6] Trademark: SUN Inc.

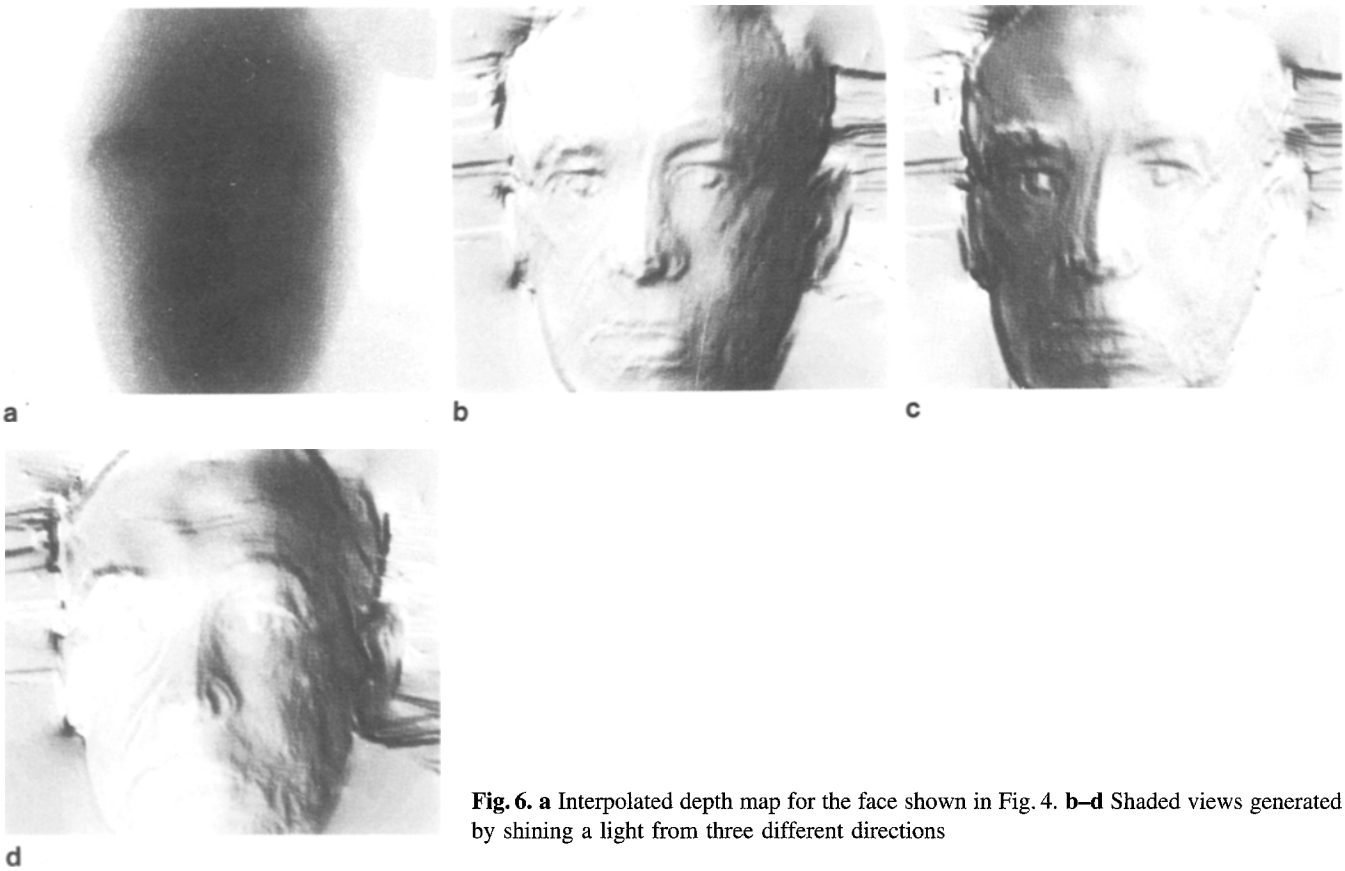


Fig. 6. a Interpolated depth map for the face shown in Fig. 4. **b–d** Shaded views generated by shining a light from three different directions

2 min 30 s on 256 × 256 images for any window size and a disparity range of 50. We are also porting this code to a multi-DSP 96002 board developed jointly by INRIA and MATRA-MSII[7] and expect run times of approximately 15 s for similar images at a cost much lower than that of a Connection Machine. Heuristics, such as a more conventional use of the hierarchy, would obviously need to be used for a faster implementation, but they are not required for any other reason than computational efficiency. Furthermore, correlation is a very regular algorithm that can be implemented in hardware (Nishihara 1984) if speed is required. 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 an internal report (Cailler et al. 1990).

In this section we have presented a 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.

[7] Funded under contract Esprit P940

Table 1. 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

Window size	256 × 256	512 × 512
3 × 3	1.75 s (79%)	5.37 s (96%)
5 × 5	3.14 s (83%)	10.13 s (96%)
7 × 7	5.03 s (85%)	17.8 s (96%)

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. 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:

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

$s = 1$ if $w0$ has been measured, 0 otherwise

$\lambda_x = 0$ if horizontal discontinuity, c_x otherwise

$\lambda_y = 0$ if vertical discontinuity, c_y otherwise

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 correlation score of Eq. 1. As shown by Szeliski (1989), 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.

$$\begin{aligned} &w0 = w + N(0, s^{-1}) \\ &\Rightarrow -\log(p(w0|w)) = 1/2\log(s) + 1/2s(w - w0)\ , \end{aligned} \tag{3}$$

and the $\partial w/\partial x$ and $\partial w/\partial x$ 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 Eq. 2 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:

$$\begin{aligned} \lambda_x &= c_x \text{Normalize}\left(\frac{\partial I}{\partial x}\right) \\ \lambda_y &= c_y \text{Normalize}\left(\frac{\partial I}{\partial y}\right) \end{aligned} \tag{4}$$

where Normalize is the piecewise linear function defined by:

$$\text{Normalize}(x) = \begin{cases} 1 & \text{if } x < x_0 \\ \frac{x_1 - x}{x_1 - x_0} & \text{if } x_0 < x < x_1\ , \\ 0 & \text{if } x_1 < x \end{cases} \tag{5}$$

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[8] 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 the Normalize function; it allows the depth information to propagate faster in the directions of least image gradient, and gives to the algorithm a behavior somewhat similar to that of adaptative diffusion schemes (e.g. Perona and Malik 1987).

[8] 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

A number of authors have investigated such regularization functionals (Blake and Zisserman 1987; Mumford and Shah 1985; Poggio 1985, to quote a few); they are roughly equivalent and we chose the quadratic criterion of Eq. 2 because it allows an extremely fast and efficient implementation on the parallel hardware that we use. We have tried to include the second derivatives in the regularizing term, with no appreciable difference except for a slowed down computation.

To compute w, we discretize the criterion of Eq. 2, yielding

$$\begin{aligned} \mathcal{C} &= \sum_{ij} s_{ij}(w_{ij} - w0_{ij})^2 + \lambda_x \sum_{ij}(w_{i+1,j} - w_{i,j})^2 \\ &\quad + \lambda_y \sum_{ij}(w_{i,j+1} - w_{i,j})^2 \\ &= S(W - W0)^t(W - W0) + W^t K W \end{aligned} \tag{6}$$

where W and $W0$ are the vectors of all w and $w0$ depths, K the sparse matrix whose "computational molecules" (Terzopoulos 1986) 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 whose elements are the correlation scores s.[9] We then use a conjugate gradient method (Szeliski 1990; Terzopoulos 1986) to solve the equation

$$\begin{aligned} &\frac{\partial \mathcal{C}}{\partial W} = 0 \\ &\Rightarrow (K + S)W = SW0\ . \end{aligned} \tag{7}$$

The parallel implementation of the conjugate gradient method involves only NEWS nearest neighbor communication and, here again, it is possible to develop specialized hardware if speed is required (Mead 1988).

In Fig. 6, we show the depth map computed by interpolating the disparity map of Fig. 4c and the 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 Fig. 7 we show the behavior 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 7b 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 Fig. 7c 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 Fig. 4 that presents few low-contrast depth discontinuities, although it

[9] $s = 0$ where the matches are invalid

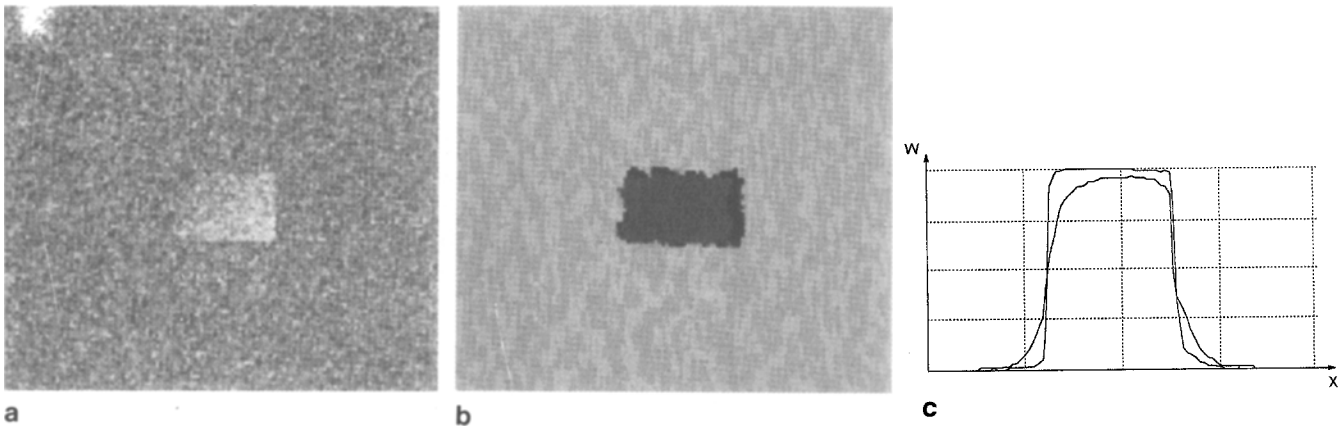


Fig. 7. a Synthetic image with a central square that presents a ramp in intensity and gaussian noise. **b** The disparity map computed by correlation. **c** A slice through a portion of the smoothed image

produces a somewhat blurry result for the tree scene of Fig. 2, as can be seen in Fig. 8a. 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 Eq. 4 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 a strong w gradient at such depth discontinuities. We take advantage of this property of our "adaptative" smoothing by defining the following iterative scheme:

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:

$$\begin{aligned} \lambda_x &= \text{Normalize}\left(\frac{\partial I}{\partial x}\right) \text{Normalize}\left(\frac{\partial w}{\partial x}\right)^{\alpha} \\ \lambda_y &= \text{Normalize}\left(\frac{\partial I}{\partial y}\right) \text{Normalize}\left(\frac{\partial w}{\partial y}\right)^{\alpha} \end{aligned} \tag{8}$$

   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 in Fig. 7c is a plot of the depth interpolated from the disparity map of Fig. 7b after four iterations. Similarly, in Fig. 8b, we show a much improved depth map for the tree scene after the same number of iterations.

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. 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 preserve 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.

Acknowledgements. The author wishes to thank Yvan Leclerc and Gerard Giraudon for their helpful comments and advice. The SUN and DSP implementations referred to in Sect. 2 were carried out by Bernard Hotz et Hervé Mathieu.

Appendix A. Behavior of the correlation algorithm on synthetic data

In this appendix we model the behavior of our correlation algorithm using synthetic data and show that the validity test defined in Sect. 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.

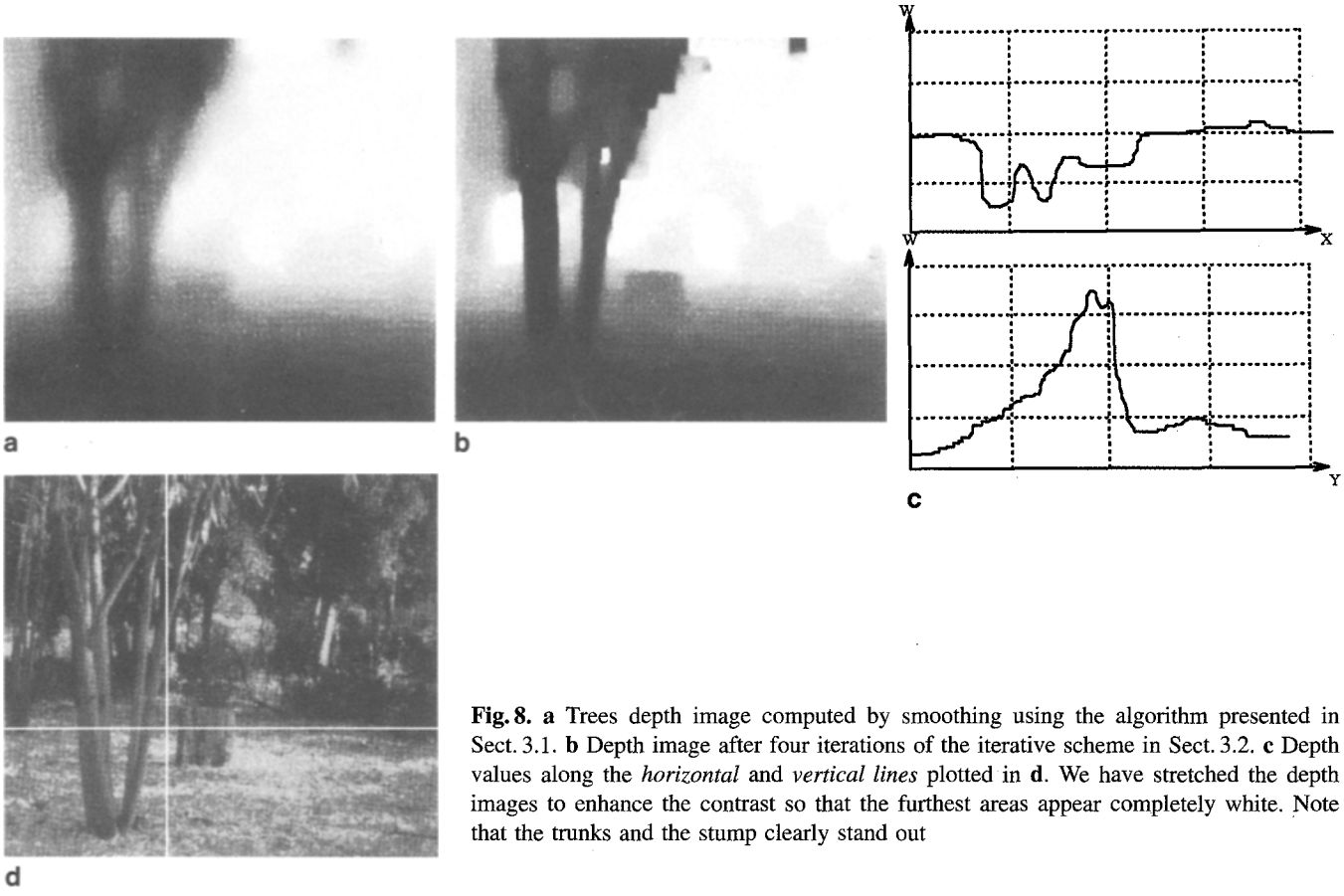


Fig. 8. **a** Trees depth image computed by smoothing using the algorithm presented in Sect. 3.1. **b** Depth image after four iterations of the iterative scheme in Sect. 3.2. **c** Depth values along the *horizontal* and *vertical lines* plotted in **d**. We have stretched the depth images to enhance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

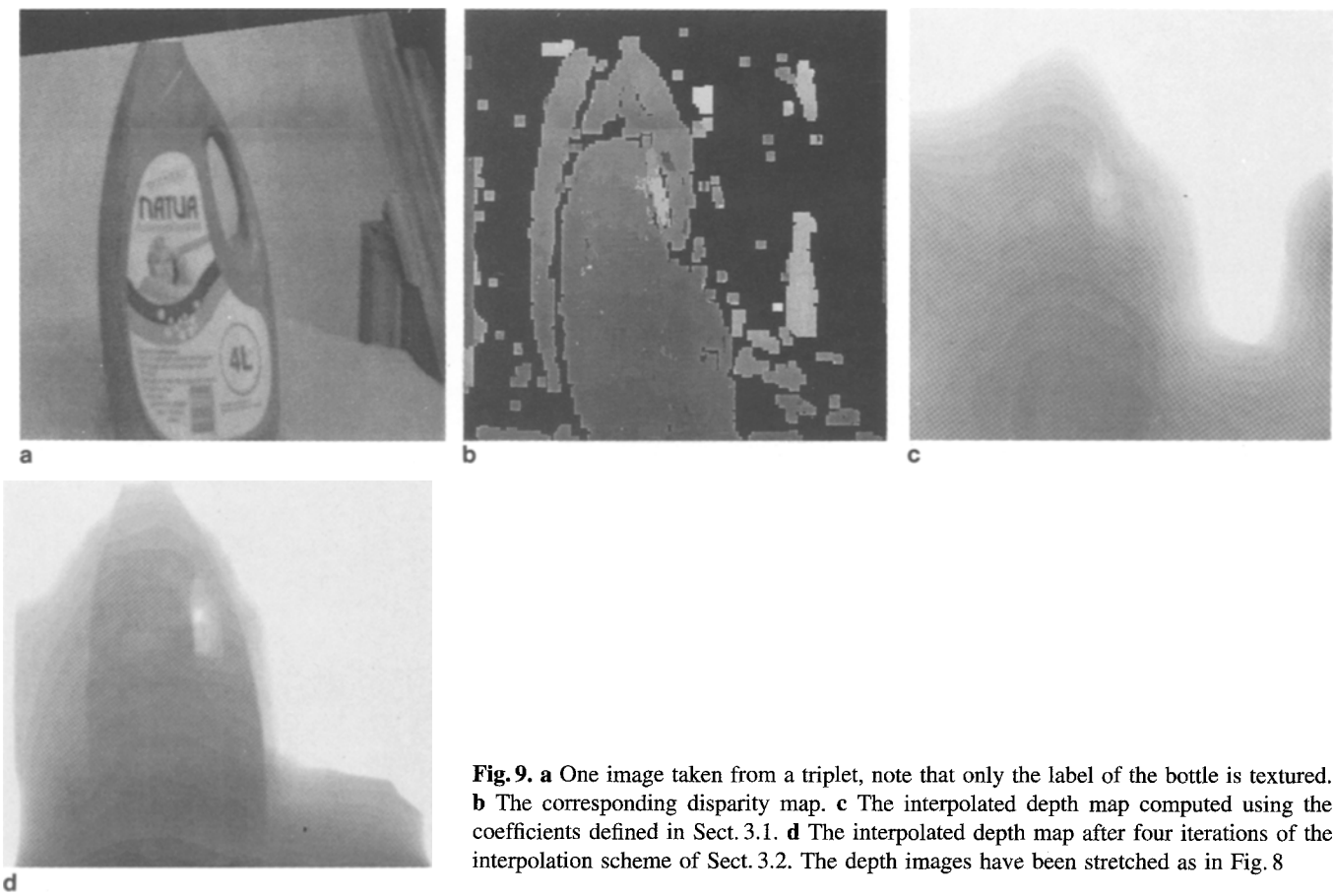


Fig. 9. **a** One image taken from a triplet, note that only the label of the bottle is textured. **b** The corresponding disparity map. **c** The interpolated depth map computed using the coefficients defined in Sect. 3.1. **d** The interpolated depth map after four iterations of the interpolation scheme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

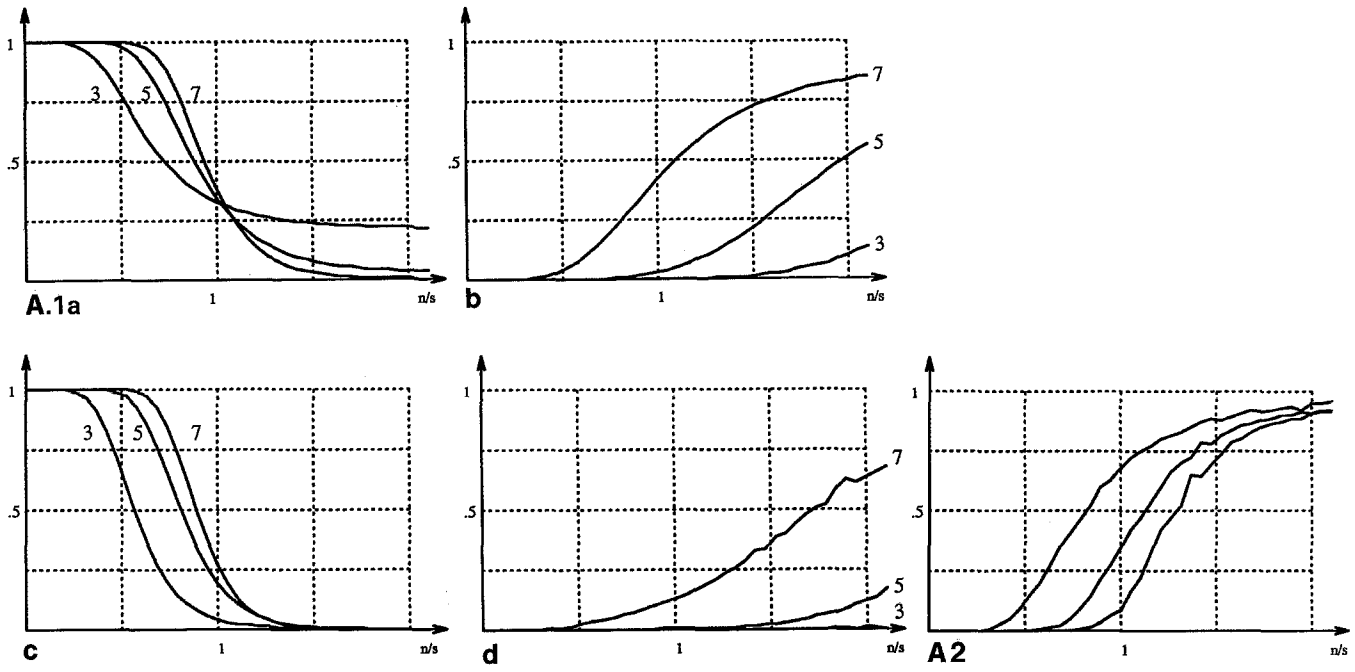


Fig. A.1. **a** The proportion of matched pixels for three window sizes, 3×3, 5×5 and 7×7, as a function of the noise-to-signal ratio. **b** The proportion of incorrectly matched pixels as a function of the noise-to-signal ratio. **c,d** The proportions of matched pixels and incorrect matches after removal of isolated matches

Fig. A.2. The proportions of pixels that are incorrectly matched when no validity test is performed for three window sizes, 3×3, 5×5 and 7×7

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

$$I_1 = N_0(0, \sigma_{\text{texture}}) + N_1(0, \sigma_{\text{noise}})$$
$$I_2 = N_0(0, \sigma_{\text{texture}}) + N_2(0, \sigma_{\text{noise}}) \quad \text{(A.1)}$$

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

$$n/s = \sigma_{\text{noise}}/\sigma_{\text{texture}} \quad \text{(A.2)}$$

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[10] 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™.

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

A.1 Influence of the noise-to-signal ratio

In Fig. A.1a 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 Fig. A.1b, 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 7×7 and 5×5 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 behavior of the two functions for 3×3 windows is fundamentally the same. However, 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, in Fig. A.2 we plot f_{error} when the correlation is performed only from I_1 to I_2 without imposing our validity criterion. f_{valid} is then computed by taking the proportion of pixels for which the score of Eq. 1 is positive. 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 to 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

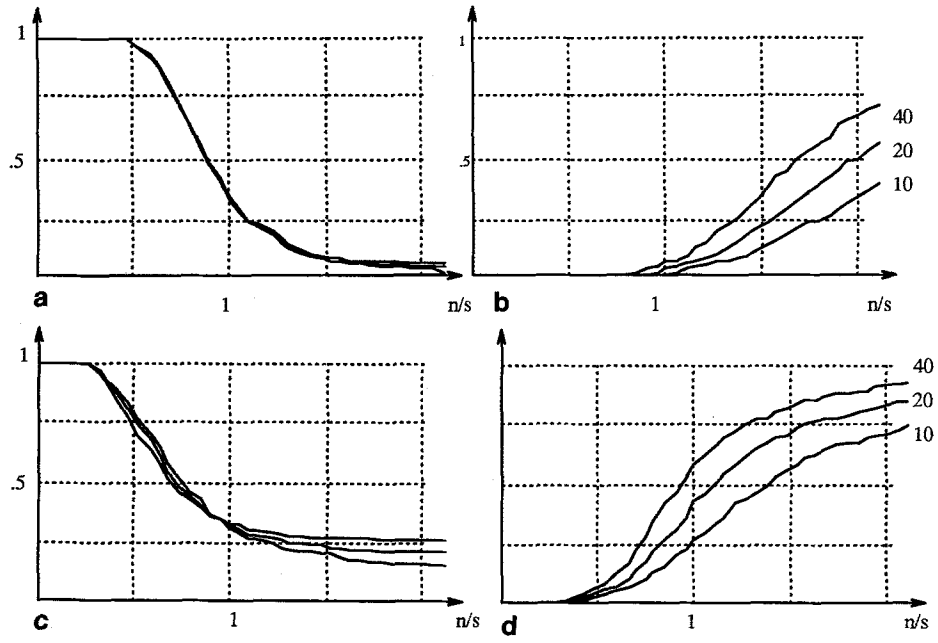


Fig. A.3a–d. The proportions of matched pixels and incorrect matches as a function of the signal-to-noise ratio for three sizes of the disparity range: **a,b** 5 × 5 windows, **c,d** 3 × 3 windows

binary array in which valid matches are represented as ones and invalid matches as zeros. We then shrink and re-expand the disparity map to remove isolated points. In Fig. A.1c and d we plot f_{valid} and f_{error} after having shrunk and re-expanded 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.

A.2 Influence of the size of the disparity range

In Fig. A.3a,b we plot f_{valid} and f_{error} computed using three sizes, 10, 20 and 40, of the disparity range and 5×5 windows. In Fig. A.3c,d we plot the same curves for 3 × 3 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.

Appendix B. Experimental comparison of the correlation measures

In this appendix, we present the experimental results obtained by Hotz and described in more detail in a technical report (Hotz 1991). In these experiments, he compared the correlation results obtained for various window sizes and the four correlation measures listed below:

$$C_1 = \frac{\sum(I_1 - I_2)^2}{\sqrt{(\sum I_1^2 \sum I_2^2)}}$$

Non-normalized mean-squared differences.

$$C_2 = \frac{\sum I_1 - I_2}{\sqrt{(\sum I_1^2 \sum I_2^2)}}$$

Non-normalized cross correlation.

$$C_3 = \frac{\sum((I_1 - \overline{I_1}) - (I_2 - \overline{I_2})^2}{\sqrt{(\sum(I^1 - \overline{I_1})^2 \sum(I_2 - \overline{I_2}^2)}}$$

Normalized mean-squared differences.

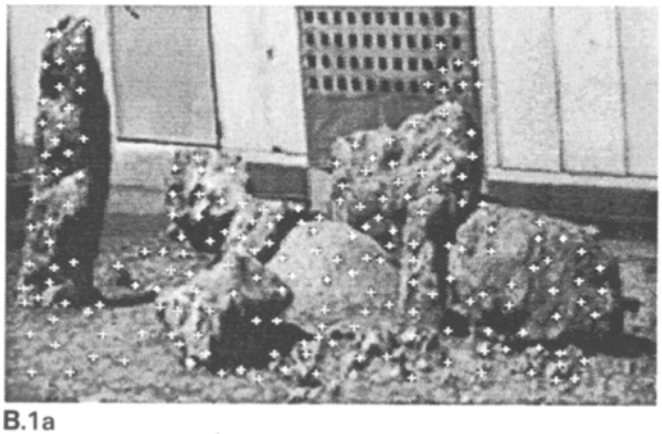
$$C_4 = \frac{\sum((I_1 - \overline{I_1})(I_2 - \overline{I_2})}{\sqrt{(\sum(I^1 - \overline{I_1})^2 \sum(I_2 - \overline{I_2}^2)}}$$

Normalized cross correlation.

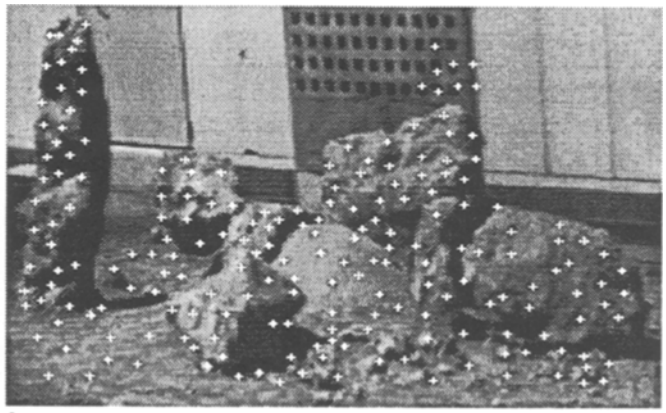
These experiments were performed on rock scenes such as the one of Fig. B.1, in which the 200 points shown as white crosses were matched by hand.[11] After the correlation procedure, the following quantities were computed:

- Percentage of the reference points for which a match has been found, which corresponds to the function f_{valid} of Appendix A.
- Percentage of the reference points for which the computed disparity is the same as the one found by hand, which corresponds to $1 - f_{\text{error}}$ where f_{error} is defined in Appendix A.

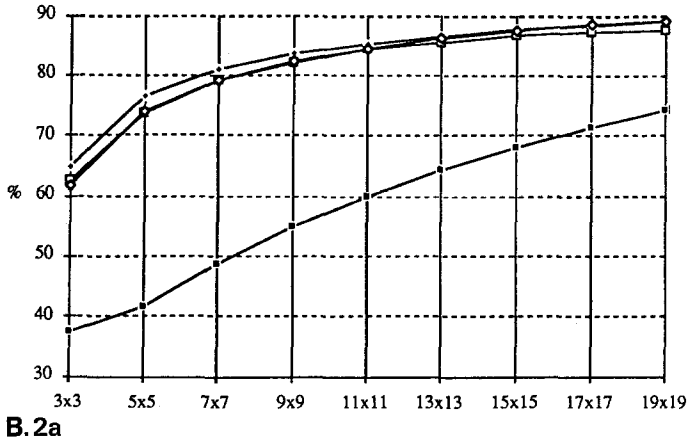
[11] Fiducial marks were first pasted on the rocks and then two images were shot for each camera position, one with the marks and one without them



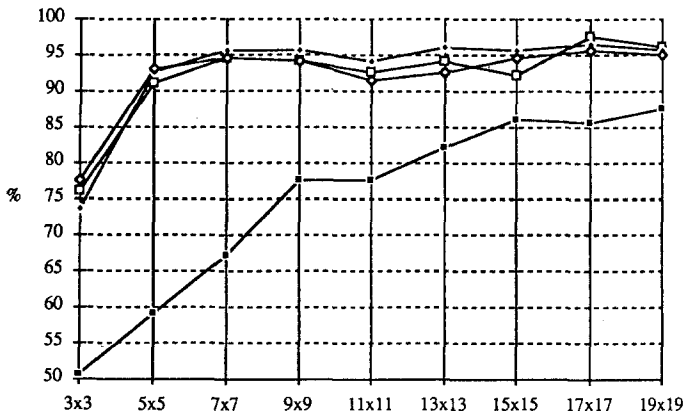
B.1a



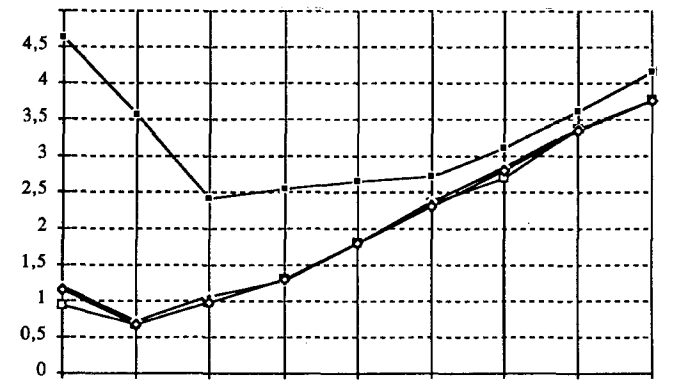
b



B.2a



b



c

Fig. B.1a,b. Two test images with 200 hand-picked matches shown as *white crosses*

Fig. B.2a–c. Results as functions of the window size for measures C_1 to C_4: **a** Percentage of reference points for which a match has been found. **b** Percentage of reference points for which the correct match has been found. **c** Average difference between the computed and "real" disparities. The four plots are superposed on the same graphs: C_1 = –■–; C_2 = –□–; C_3 = –♦–; C_4 = –◇–

– Average difference between the computed disparities and the hand-picked ones.

in Fig. B.2 and for each of the correlation measures C_1 to C_4, we plot the results as functions of the window size. The plots all have essentially the same shape: the percentage of matched points increases with the window size while the precision decreases. By using large windows, we smooth out the finer details and, in effect, reduce the resolution.

Scores C_2, C_3, and C_4 yield results that are very similar and are much better than the ones computed using score C_1. It is easy to understand why in the case of C_3 and C_4; both criteria being normalized, they are insensitive to variations in the mean intensity value of the images that can overwhelm criterion C_1. It is more interesting to note that C_2, even though it is not normalized, is also much better than C_1.

By changing the camera settings for one image of the stereo pair, we have checked that the normalized criteria are effectively insensitive to such transformations while C_2 degrades somewhat and C_1 degrades dramatically.

For a more exhaustive description of these tests, we refer the interested[12] reader to the technical report (Hotz 1991).

[12] and francophone

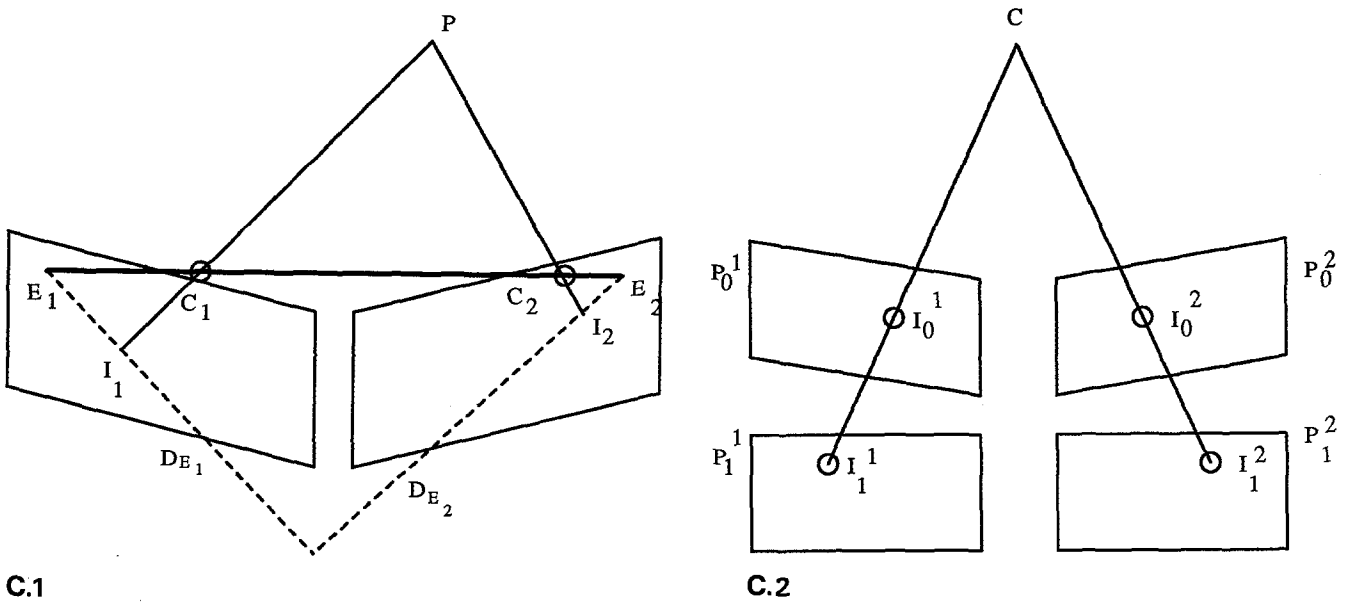


Fig. C.1. Pinhole model for two cameras: C_1 and C_2 are the optical centers of the two cameras and the world point P projects to I_1 and I_2 respectively. E_1 and E_2 are the epipoles through which all epipolar lines go, and D_{E_1} and D_{E_2} are the epipolar lines on which I_1 and I_2 are bound to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

Fig. C.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 image plane $P_1^1 = P_1^2$

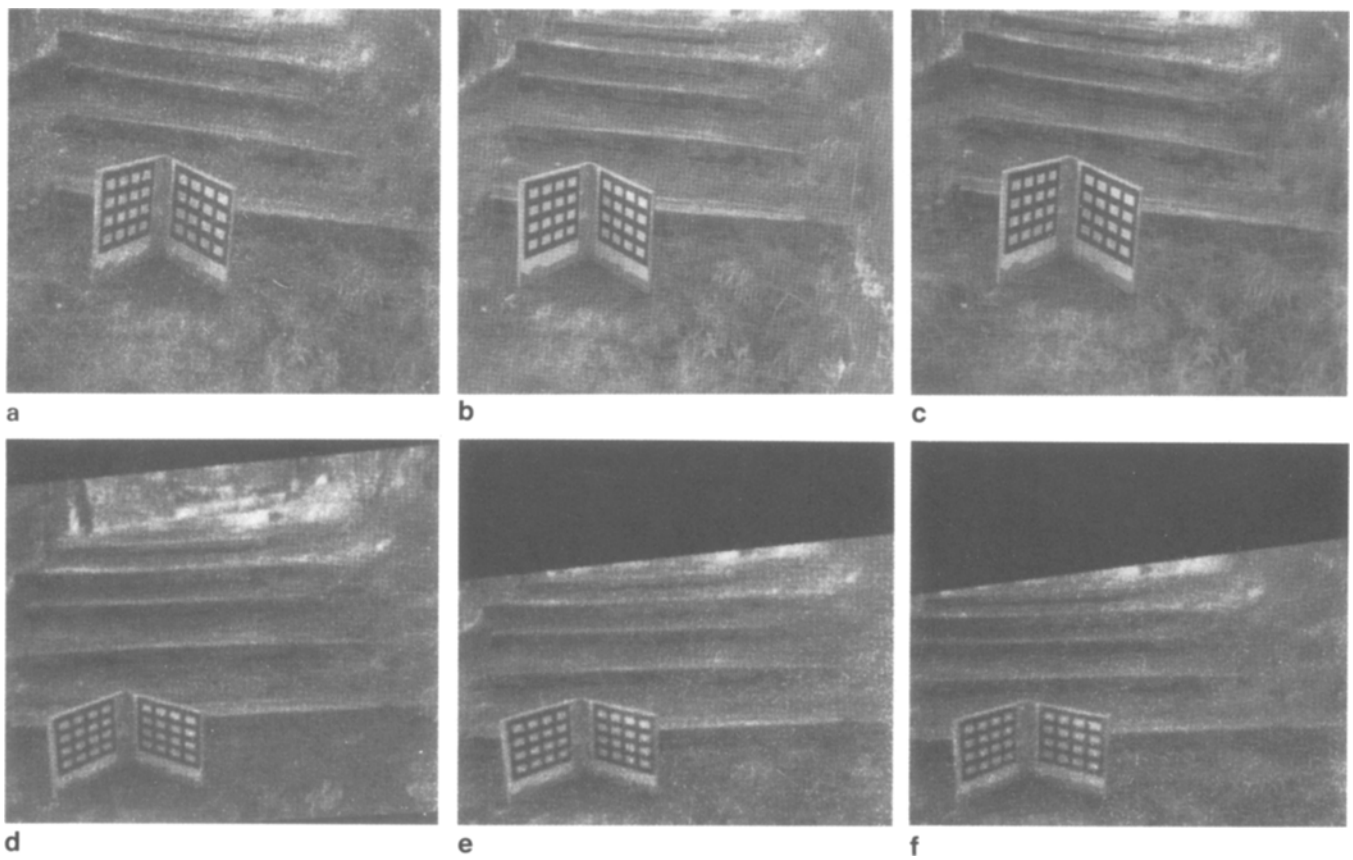


Fig. C.3. a–c A triplet of images. **d–f** The images after rectification. The grid in the bottom left corner is used to calibrate the system and compute the T matrices

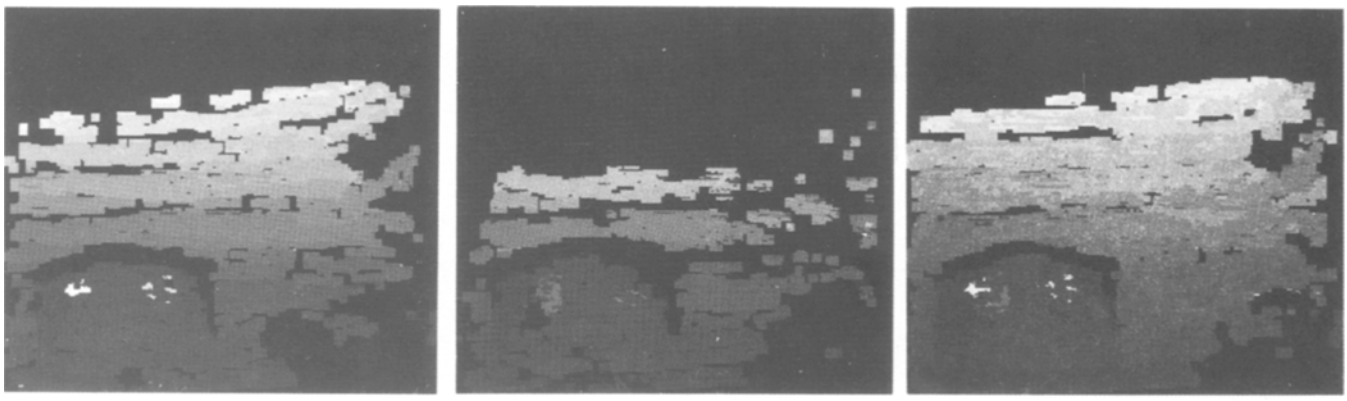


Fig. C.4a–c. Disparity maps: **a** computed by correlating Fig. C.3a with C.3b, **b** by correlating C.3a with C.3c, and **c** merging the two maps. These maps are virtually error-free except for those caused by the repetitive patterns on the grid itself

Appendix C. Rectification

In this appendix, we describe the rectification techniques we use to deal with triplet 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 and Hansen (1988).

C.1 From image planes to calibration matrices

Each camera is modeled, using the classic pinhole model, by its optical center $\mathcal{C}$, its image plane $\mathcal{P}$ and a 4×3 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:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = T \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{C.1}$$

$u = U/W$

$v = V/W$

T is such that

$$T\mathcal{C} = 0 \,. \tag{C.2}$$

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

$$ax + by + zc = h \quad \text{where} \quad a^2 + b^2 + c^2 = 1 \tag{C.3}$$

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

$$M_0 = \begin{pmatrix} h & 0 & 0 & a \\ 0 & h & 0 & b \\ 0 & 0 & h & b \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{C.4}$$

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

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 $\overrightarrow{\mathcal{OC}}$. Using the pinhole camera model, it is easy to see that the matrix

$$M = T_0 M_0 T_0^{-1} \tag{C.5}$$

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 Fig. C.1. The projective image coordinates of I can be computed from its world coordinates by multiplying them by a 3×4 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

$$T = NM \,. \tag{C.6}$$

C.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. 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 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 Fig. C.2. To every image point of the original image plane corresponds a unique point of the rectified plane; we derive below their analytical relationship.

Let $T_0 = [R_0, C_0]$ and $T_1([R_1, C_1]$ be the two corresponding 3×4 calibration matrices computed as described above, where R_0 and R_1 are 3×3 matrices and C_0 and C_1 3×1 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 point in the rectified image. Let then P be a world point that projects at both I_0 and I_1, i.e.

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

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

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

we can write

$$\begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} = \lambda T_0 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \lambda \left(R_0 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + C_0 \right)$$

$$\begin{pmatrix} U_1 \\ V_1 \\ W_1 \end{pmatrix} = \lambda T_1 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \lambda \left(R_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} + C_1 \right)$$

$$\Rightarrow \begin{pmatrix} U_1 \\ V_1 \\ W_1 \end{pmatrix} = R_1 R_0^{-1} \left(\begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} - C_0 \right) + wC_1$$

$$= R_1 R_0^{-1} \begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} + (C_1 - R_1 R_0^{-1} C_0)$$

$$= \text{Rect} \begin{pmatrix} U_0 \\ V_0 \\ 1 \end{pmatrix} \tag{C.9}$$

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

C. 3 Rectifying the images

Having computed the rectification matrix Rect of equation C.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 $I_0 = \text{Rect}^{-1} I_1$ in the original image and we compute the grey level of I_1 using bilinear interpolation. In Fig. C.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 (Toscani et al. 1989). Note that the rectified images are not very deformed because the three original image planes were almost parallel. In Fig. C.4 we show the output of our stereo algorithm.

References

Anandan P (1989) A computational framework and an algorithm for the measurement of motion. Int J Computer Vision 2(3):283–310

Ayache N, Hansen C (1988) Rectification of images for binocular and trinocular stereovision. Ninth International Conference on Pattern Recognition. Rome, Italy, November 1988, pp 11–16

Ayache N, Lustman F (1987) Fast and reliable passive trinocular stereovision. First International Conference on Computer Vision

Barnard ST, Fischler MA (1982) Computational stereo. Comput Surv 14(4):553–572

Blake A, Zisserman A (1987) Visual Reconstruction. MIT Press, Cambridge, MA

Brown CM, Ballard DH (1982) Computer Vision. Prentice-Hall, Englewood Cliffs

Burt PJ, Yen C, Xu X (1982) Local correlation measures for motion analysis. In IEEE PRIP Conference, pp 269–274

Cailler C, Fornarix F-X, Heng P, Holtzer T (1990) Cocosun. Rapport de stage, Cerics

Faugeras OD (1988) A few steps toward artifical 3D vision. (Rapport de Recherche 790) INRIA, Sophia-Antipolis

Güelch E (1988) Results of test on image matching of isprs wg iii. Fourth Int Arch Photogrammetry Remote Sensing 27(III):254–271, and accompanying poster presentation

Hannah MJ (1988) Digital stereo image matching techniques. Int Arch Photogrammetry Remote Sensing 27(III): 280–293

Hotz B (1991) Etude de techniques de stéréovision par corrélation. (Rapport des stage de dea) CNES, Toulouse, France

Kanade T, Okutomi M (1990) A stereo matching algorithm with an adaptative window: theory and experiment. Image Understanding Workshop

Mead C (1988) Analog vlsi for auditory and vision signal processing. INSPEC Conference, San Francisco, Calif

Meygret A, Thonnat M, Berthod M (1990) A pyramidal stereovision algorithm based on contour chain points. ECCV90 Conference, Antibes

Moravec H (1981) Robot visual navigation. UMI Research Press, Ann Arbor, MI

Mumford J, Shah D (1985) Boundary detection by minimizing functionals. CVPR85 Conference, San Francisco, Calif, June 1985, pp 22–28

Nishihara HK (1984) Practical real-time imaging stereo matcher. Optical Eng 23(5)

Nishihara HK, Poggio T (1983) Stereo vision for robotics. ISRR83 Conference, Bretton Woods, NH

Perona P, Malik J (1987) Scale space and edge detection using anisotropic diffusion. IEEE Computer Society Workshop on Computer Vision, Miami, Fl, pp 16–22

Poggio T, Torre V, Koch C (1985) Computational vision and regularization theory. Nature 317:314–319

Szeliski R (1989) Bayesian modeling of uncertainty in low-level vision. Kluwer Academic Press, Norwell MA

Szeliski R (1990) Fast surface interpolation using hierarchical basis functions. IEEE Trans Pattern Analysis Machine Intelligence 12(6):513–528

Terzopoulos D (1986) Image analysis multigrid relaxation methods. IEEE Trans Pattern Analysis Machine Intelligence 8(2):129–139

Toscani G, Vaillant R, Deriche R, Faugeras OD (1989) Stereo camera calibration using the environment. 6th Scandinavian Conference on Image Analysis. pp 953–960