Enhancing the Volumetric Approach to Stereo Matching

ITALO DE OLIVEIRA MATIAS[1], ANTONIO A. F. OLIVEIRA[2], LUIZ M. G. GONÇALVES[3]

[1]Rede Globo
Rua Von Martius 22, Cep 22460-040, Rio de Janeiro, Brasil
italo.matias@tvglobo.com.br
[2]Universidade Federal do Rio de Janeiro
CP 68511, 21945-970, Rio de Janeiro, Brasil
oliveira@lcg.ufrj.br
[3]Universidade Estadual de Campinas
CP 6172, 13083-970, Campinas, SP, Brasil
lmarcos@ic.unicamp.br

Abstract. We propose techniques to enhance the volumetric approach to stereo matching [9, 10, 11], with which to obtain dense disparity and detect occluded zones. The volumetric approach works on the $row \times column \times disparity$ space initializing the voxels with a measure of the similarity between the stereo pair that they represent (Similarity Phase). In this phase, we propose a function L_0 that provides a better initial estimation for disparity. Then, these values are refined through an iterative process which inhibits all but one voxels placed along the same line of sight (Competition Phase). In this phase, we propose to use a Dynamical Programming technique for faster converging to the final solution and producing smoother maps. We substantially reduce the proportionality constant of the original algorithm complexity, enhancing time without strongly influencing the results.

1 Introduction

Reconstruction of "shape from stereo" consists on the determination of depth from two (or more) images of (almost) the same scene obtained from imaging sensors with different spatial positioning. Of course, the images should have a common covering area (the "almost" meaning). The fundamental basis and also the bottle-neck for stereo reconstruction, a problem that remains until today, is the matching process. The matching result is known as the disparity map, that is, a map containing, for each pixel in one image, the displacement in image coordinates to the corresponding pixel in the other image. In general, it is desired for a stereo matching algorithm to produce a dense disparity map, also smooth and detailed. The Marr and Poggio [5] paradigm states the uniqueness of disparity and its continuity within a segment, that is, each position of a disparity map has a unique value defined (this somewhat depends on the opacity of the scene) and the values along the map are continuous (almost everywhere). In the volumetric approach to stereo matching proposed in [9, 10, 11] (from now on we cite only [11]), uniqueness is obtained through a competition process between all voxels that project onto a same pixel. Smoothness is improved by using a smoothing filter in the evaluation of iterative functions employed in the competition process. In the current work, we propose ways to enhance the volumetric approach as proposed in [11]. In the same way, we try to find regions in which disparity can be well defined, then to use an iterative function to generate correct matchings. The two hypothesis (uniqueness and continuity) are used in the same 3D space ($row \times column \times disparity$), but with a different strategy for faster converging. We basically introduce two contributions: a better function for estimating initial disparities and the use of a Dynamical Programming (DP) technique to get faster and smoother solutions. As we will demonstrate, the algorithm presented here takes a substantially smaller time than the one presented in [11] while maintaining coherence and precision in the determination of occluded zones. A comparison of both approaches performed in number of iterations shows that we get an algorithm almost five times faster than the original one [11]. Also, it will be shown that the lost in precision is very small, what can be used to justify the gain in time.

2 Related works

In the last decades, several works were guided by the clear perception of completely filled surfaces noted when one looks to random dot stereograms, even if the points are sparse. Julesz [3] used stereograms containing acute edges (with big depth differences along them), conjecturing that, in this case, the reconstruction process is performed at a very fine resolution. A substantial amount of works followed this approach, trying to produce a dense disparity map from fewer determinations of disparity in well defined positions as corners and edges. It is not clear from those experiments that humans have a completely filled representation. Also, this is not the unique approach to stereo match-

1530-1834/01 $10.00 © 2001 IEEE

ing problem. Nishihara [7, 6] adopted an opposite position. He concluded that humans has a, surprisingly, poor spatial resolution, however a great tolerance to noise. In his model, filled perception of acute edges can be explained as an illusion created by the presence of frontiers with different luminance. It seems that a flag of consistency indicates the presence of a filled surface. It is not necessary a filled representation to justify human performance. This assumption reinforces the idea of matching in sparse positions, using area based methods, resulting in a minimum measurement tool [6]. Note that in certain cases, these minimum measurements would be enough (as in robotics).

In general, it is not trivial to establish a complete (dense) disparity map, due to occlusions, systematic errors and also depending on the method used to calculate disparity (e.g, "element" based methods determine a sparse map). In practice, the matching process can be used to determine the maximum as possible number of correct matches and then other methods (as relaxation) used to spread (or interpolate) disparity to other points. The algorithm offered by Zitnick and Kanade [11] follows this approach. Basically, a raw map calculated by a simple correlation approach is used as the starting point for their algorithm. We note that this initial map should provide a good estimation for the matching confidence, mainly in non occluded zones. In general, this is not trivial because of problems inherent to the existing correlation approaches. From this initial map, a relaxation rule is used to spread correct matches to other positions, finding a complete map with explicit detection of occluded zones. The time spent in this iterative phase to generate the final disparity map is the main problem with their algorithm. It is reported in [11] that the algorithm takes around 15 iterations. In this way, our proposal is to introduce improvements in that time, with a minimum (acceptable) lost in precision, carrying out an efficient and robust approach. Some experiments demonstrate that the modified version takes some few (3 to 7) seconds in the whole process (raw map plus 2 or 3 iterations to get the final map) for images of 300 × 300 pixels.

3 Understanding the LCD space

In order to better understand the LCD coordinate system, we present some concepts and the basic idea on how to represent a line in this system. When working in XYZ coordinate system, in general, the origin lies in the center of projection of left image. In the LCD system, each center of projection lies on the vertical line passing in the images center. Considering that this center has coordinates $(X/2, Y/2)$, these values can be used to establish relations between x, y, l and c. The voxels of the cube LCD having coordinates $(l = L/2, c = C/2, d)$ project in the left image center. Using an ideal model for stereopsis (Figure 1), where f is the focal distance, b is the baseline, z is depth, L the displacement in X (no Y displacements), it is possible to rewrite the equations of a straight line from LCD to XYZ system (and vice-verse). Let r an oblique straight line described in XYZ system and let k its corresponding in LCD system as seen in Figure 2. Figure 3 shows that an oblique line represented as a raster may intercept more than one pixel in a same column. In this way, the line $(l, c) = k$ expressed in XYZ system, has parametric equation given by:

$$\left(\frac{z}{f}\left(\frac{X}{f} - K_c\right) + \frac{L}{f}, K_l - \frac{Y}{f} + \frac{C}{f}, z\right) \quad (1)$$

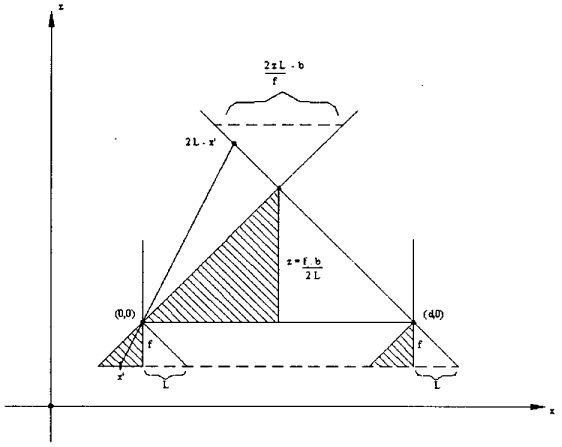

Figure 1: An ideal model for stereo

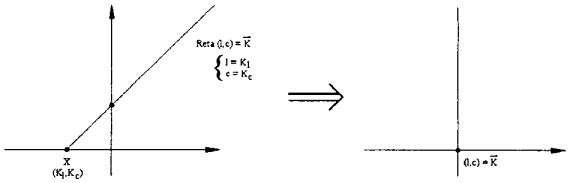

Figure 2: Straight line in LCD system

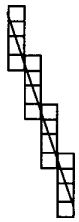

Figure 3: Raster representation of a straight line

Considering stereopsis Equations [1], the line where a point (k_l, k_c, k_d) lies is given (in XYZ) as:

$$\left(\frac{z}{f}\left(\frac{X}{2} - \left(K_c + K_d\right) + b + \frac{L}{2}, K_l - \frac{Y}{2} + \frac{C}{2}, z\right)\right) \quad (2)$$

Note that c and x relates to each other by:

$$x = \frac{b}{d}\left(\frac{X}{2} - c\right) + \frac{L}{2} \quad (3)$$

So, in LCD system, the Equation of a straight line can be given by:

$$(c + d) = (k_c + k_d) \quad (4)$$

4 The (cooperative) volumetric approach

In the volumetric approach to stereo, given two images in a 2D space ($L \times C$), disparity is considered as the third dimension, constrained in a sub-space of dimension D, that is, $d = 0, ..., d_{max}$. Without loss of generality, it can be assumed that the images are rectangles. Each voxel (l, c, d) from the space LCD projects on the pixel (l, c) in the left image and on the pixel $(l, c+d)$ in the right image. A function defined in the LCD space is used to refine the disparity values. Let $L_n(l, c, d)$ this updating function in a given iteration n and its value attributed to the voxel (l, c, d). Initial values for $L_0(l, c, d)$ can be calculated using any function that measures similarity as for example normalized correlation used in [11]. Ideally, this function should produce an initial disparity map with confident (high) values for the matchings determined for actually corresponding pixels. Besides, note that what we also hope for the opposite may not occur, that is, many non corresponding points will have a high similarity value, given by the above function. This depends on the images, on how to choose parameters as the local support and search window dimensions, and also on system acquiring errors. Further, it is necessary to normalize the above values of L_0. Here we introduce the first improvement over the original algorithm. We use a simple summation of absolute differences (*SAD*) function as L_0 and a different normalization function, given by:

$$L_0(l, c, d) = \frac{I_{max}}{L_0(l, c, d) + I_{max}} \quad (5)$$

where I_{max} is the maximum intensity (gray level) value of the image (that is, 255). In practice, this normalization produces values in the interval [0.1, 1.0]. Values close to 1.0 means a good match, while values close to 0.1 can be discarded from next iterations. In the experiments performed in this work, we get good results by setting the parameters of L_0 to give at least 40 % of initially correct values of similarity. The practical result of applying L_0 can be seen in Figure 4. Each cut plane is parallel to the planes formed by the rows and columns that contain the result of the similarity function $L_0(l, c, d)$ in that plane. The value of disparity is kept constant for all voxels in the same plane, that is, the first plane corresponds to $d = 0$, the second to $d = 1$, and so on until $d = D_{max} - 1$. Ideally, for a given pixel (l, c), the values of $L_0(l, c, d)$ are defined in a unique

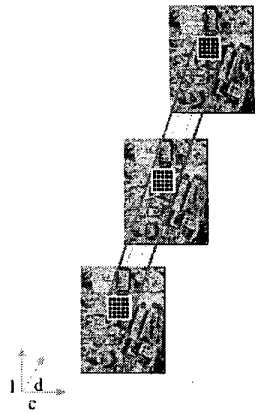

Figure 4: L_0 for a given pixel (l, c) in the 3 first planes $(d = 0, d = 1, d = 2)$ of LCD space.

plane $d = constant$ in which similarity is maximum. As said above, in practice, that does not happen due to errors and occlusions. The volumetric algorithm try just to discard these false matchings using the unicity hypothesis.

4.1 The local support

The continuity hypothesis implies that neighboring voxels have consistent similarity values. In [11], an average of neighboring values in a 3D space is used to augment consistency. The average is calculated inside a volumetric region called local support (5). It determines what neighboring voxels should contribute to the average. Actually, the local support should include only voxels that may contribute with correct similarity values. The local support dimensions influence in the final results because it averages neighboring values for a given voxel. Note that a huge local support could include wrong values as well as a small one. Several approaches were proposed using 3D support (see [2, 8, 4]). In [4], a detailed analysis of the several relations including their main differences is presented. Here, as in [11], we use a fixed size for the local support. We get good results with a local support of $5 \times 5 \times 3$, that is, 5 voxels in L, by 5 in C by 3 in D. In [11], the value $S_n(l, c, d)$ of the local support for each voxel (l, c, d) of the LCD space is given by the summation of all similarity values in a region Q as:

$$S_n(l, c, d) = \sum_{(l', c', d') \in Q} L_n\left(l + l', c + c', d + d'\right)) \quad (6)$$

4.2 Competition between voxels

The uniqueness hypothesis implies that only a matching occurs for each pixel. Let $P(l, c, d)$ the inhibition zone, that is, the set of voxels superposed to an element (l, c, d) when projected in the image (remember each voxel in $P(l, c, d)$ projects in the pixel (l, c) in the left image and in the pixel

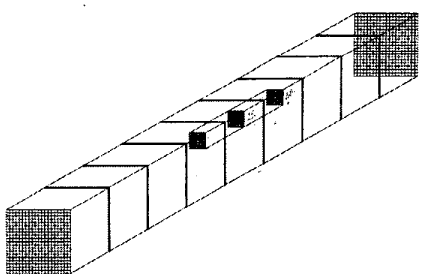

Figure 5: Local support

$(l, c+d)$ in the right image). As given in [11], the average $R_n(l,c,d)$ resulting from inhibition in elements of $S_n(l,c,d)$ in function of the elements in $P(l,c,d)$, is:

$$R_n(l,c,d) = \left(\frac{S_n(l,c,d)}{\sum_{(l',c',d') \in P} S_n(l',c',d')} \right)^{\alpha} \quad (7)$$

Also, in [11] the match value relative to the image similarity is constrained by the pixel (l,c) in the left image and by the pixel $(l,c+d)$ in the right image, given by:

$$T_n(l,c,d) = L_0(l,c,d) * R_n(l,c,d) \quad (8)$$

So, the final updating function, as given in [11], is a combination of equations 6,7,8:

$$L_{n+1}(l,c,d) = L_0(l,c,d) * \left(\frac{S_n(l,c,d)}{\sum_{(l',c',d') \in P} S_n(l',c',d')} \right)^{\alpha} \quad (9)$$

Based on the initial proposal of Marr and Poggio, in [11] Equation 9 is rewritten as:

$$L_{n+1}(l,c,d) = \sigma \left(S_n(l,c,d) - \epsilon \sum_{(l',c',d') \in P} S_n(l',c',d') \right) \quad (10)$$

where σ is the sigmoid function and ϵ is the inhibition constant.

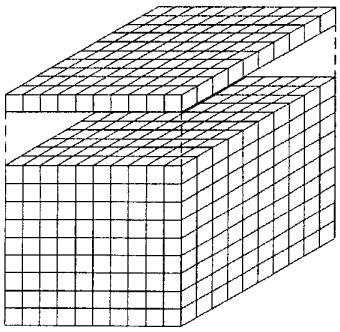

Figure 6: Taking slice $l = 0$ for DP.

5 Enhancing the volumetric algorithm with DP

The use of DP to improves the algorithm convergence and also produces a smoother map. Basically, the cube LCD is divided into slices having L constant, as seen in Figure 6 for $l = 0$. Over each plane of line $l = constant$, we compute a minimum cost function (MCF) taking into account the similarity value. DP explores the smoothness and continuity restrictions assumed for the disparity map, avoiding the creation of holes in the resulting map. The 3D problem is broken into sub-problems in the 2D space. In each slice, we try to keep differentiability of disparity from a pixel to its neighbor. That is, we consider that $L_{n+1}(l,c,d)$ and $L_{n+1}(l,c+1,d)$ varies smoothly or, given a limit ϵ for smoothness:

$$|L_{n+1}(l,c,d) - L_{n+1}(l,c+1,d)| < \epsilon \quad (11)$$

Figure 7 exibit the optimal path of disparity in a 2D slice. To find this path, the following methodology is applied:

- determine the voxels which have the highest values of $L_{n+1}(l,c,d)$, in a certain column;
- calculate the minimum cost function from columns $c = 1$ until $c = C$;
- in each column, voxel (l,c,d) chosen determines the pixel (l,c) in the resulting map.

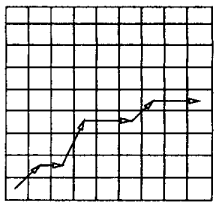

Figure 7: Minimum cost function on slice $l = K$.

5.1 The complete algorithm and its complexity

By introducing the above technique in the original approach, the whole algorithm for determining the dense disparity can be summarized as:

1. Prepare a 3D array and set initial values of similarity $L_0(l,c,d)$ for each voxel (l,c,d) (Equation 5);
2. iteratively update the values for $L_{n+1}(l,c,d)$ using expression 9, until all voxels satisfy the termination rule; voxels whose matching values get too low are no more updated;
3. use DP over the values $L_{n+1}(l,c,d)$,in a network including only the voxels with highest values that determine a voxel $V^*(l,c,d)$ in the 3D array corresponding to pixel (l,c);
4. if the matching value for voxel V^* is too low, classify the corresponding pixel as being occluded; otherwise, its output generates the disparity d for that pixel.

The complexity of the above algorithm is N^2DI, where N^2 is the image size, D is the maximum interval for disparity adopted and I is the number of iterations performed. The size of required memory is N^2D. In practice, the proposed modification makes the algorithm substantially faster than the algorithm proposed in [11]. In the experiments, the algorithm converged to final values with 2 iterations in average (one experiment had 3 iterations). This time is some 5 to 7 times faster than the original one [11], even considering the cost for calculating the MCF in each slice.

5.2 Occlusion detection

We identify occlusion by applying the unicity restriction on the final values in the disparity map. The function proposed for calculating the initial values for $L_0(l, c, d)$ allows substantial elimination of false correspondences. Note that if no false matchings occur (low similarity value for occluded pixels) the algorithm works better. Thus, in some images the threshold for a pixel to be occluded could be a little higher, but to attend the general case, we consider this threshold as being 0.11. We remark that the initial values in L_0 gives already a good prediction if a pixel tends to be occluded. So, in general, occlusion areas are represented in the function $L_{n+1}(l, c, d)$ as regions having intensity values not similar in the interval D of disparity considered. So, after the similarity values being converged, one can determine if a pixel is occluded by trying to find an element with greater value in its line of site. In this way, we note that the above DP technique allows the detection of occluded zones. Besides, the main objective of its use is to accelerate processing time and produce a smoother final map.

6 Experiments and results

The main proposal in [11] was the explicit detection of occlusion. Our main proposal is to accelerate the whole process while maintaining a certain coherence in occlusion detection and to get a better suavization of the final map. In this section we show results and compare them with Zitnick's algorithm considering these aspects. We used a PC with a Pentium II processor (300 Mhz), with "OpenGL" plus "C" language for graphical interfaces. The images used were gently offered by Zitnick from [9, 10, 11] and by Gonçalves [1]. We selected three distinct pairs to visualize our results. The occlusion threshold was set around 0.111 and the convergence factor (σ) varied from 1.01 to 1.05 in the experiments. In almost all tests, the factor α has not influenced the convergence because of using DP, which was used in all iterations from L_1 to L_n (highest n was 3).

Figure 8 shows the coal mine pair. Initially, we used a local support of $5 \times 5 \times 3$, and a window size of 3×3 for determination of L_0 (left of Figure 9). By using DP, in only two iterations we got the result shown in right side of Figure

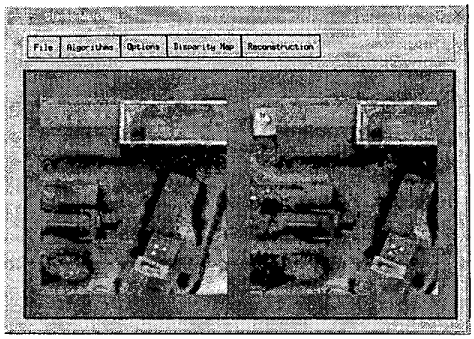

Figure 8: Coal mine pair.

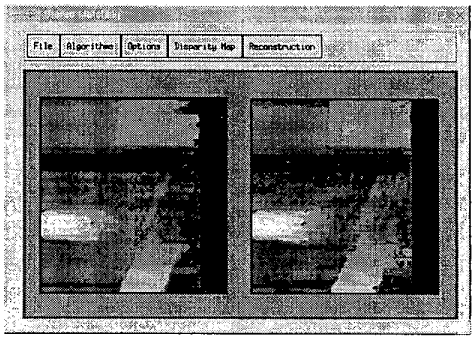

Figure 9: L_0 on the left, L_2 on the right.

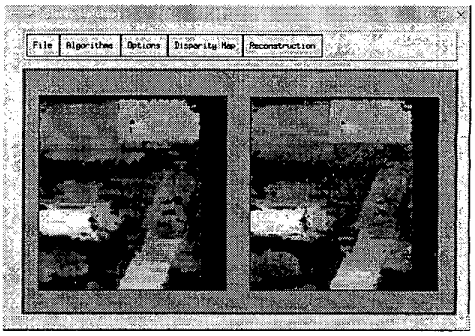

Figure 10: L_0 on the left, L_2 on the right.

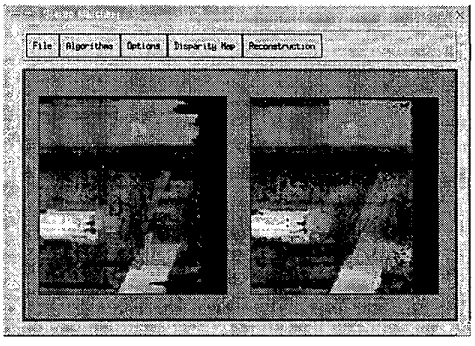

Figure 11: L_0 on the left, L_2 on the right. (Local support $3 \times 3 \times 3$.

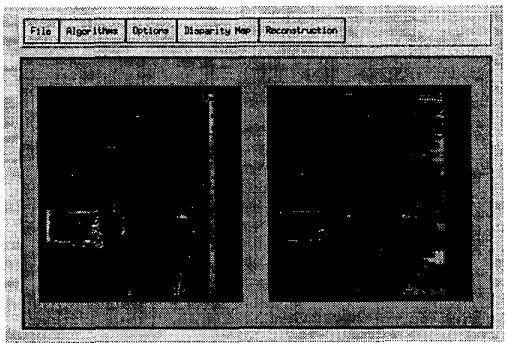

Figure 12: Difference between two iterations.

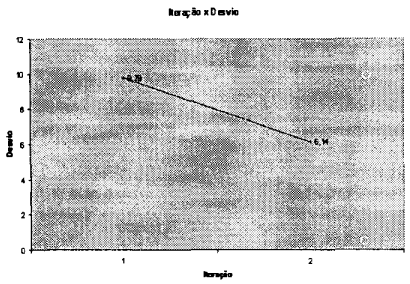

Figure 13: Deviation × iteration difference.

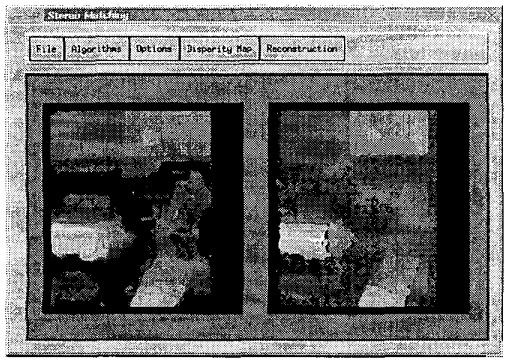

Figure 14: Comparison with ideal result.

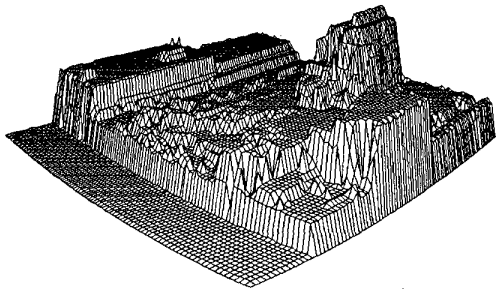

Figure 15: Reconstruction from disparity map.

9, very close to the one shown in [10], and, in this particular experiment, 7 times faster (in number of iterations). We got L_0 in 1.93 seconds and the final result in 6.91 seconds. For the MCF used in DP, we used a cut point of 0.75 for similarity values. Here, the DP technique, besides accelerating the process, also presents a smoothing effect producing a disparity map with noted continuity from point to point.

The process can yet be accelerated by augmenting the cut point for the DP, but this could loose details and generate false matchings, concequently false oclusions. Figure 10 illustrates this effect. The left image is L_0 and the right image is the final map L_n. The local support and window size are the same as above and the number of iterations (n) is only two. The time spent was 1.93 sec for determination of L_0 and (counting L_0 time) 5.76 sec for L_2. The cut point for DP was 0.93. The total time spent is more than one second faster, but we can see the presence of false matchings and false occlusions. Another way to speed up the process is decreasing the dimensions of the local support. In Figure 11, the local support was reduced to $3 \times 3 \times 3$. The time for calculating L_0 was 1.93 sec. Again, with two iterations and overall of 5.02 sec, we got the result shown in right side, with a cut point of 0.90 for faster converging. We gain in time by decreasing local support dimensions, but one can see that appears false matchings and false occlusions.

In Figure 12, we show variations from iteration L_1 to L_0 (left image) and from L_2 to L_1 (right image) as a means to see the gain in convergence. We calculated the standard deviation from the above maps as being 9.79 in the first one and 6.14 in the next, shown in Figure 13. This illustrates the fast convergence of the algorithm to final values, once we have fast decreasing deviations. Following, we compare our algorithm (left of Figure 14)with an ideal result (right), also used in [10]. It can be explicitly noted the occlusion zones, given by the small dark regions. In this case, we used a window size of 9×9 for determination of L_0 and further a local support of $7 \times 7 \times 3$. With only 3 iterations, we got the result shown in the right side of Figure 14, with a total time of 13.02 sec. We used 0.88 as the cut point for the DP convergence. We can see that the final result is very close to the ideal one. We note that appears some false matchings due to the cut point used to calculate the MCF. It is valid to comment that the significant increasing on time (13 sec) is due to the augment in the local support dimensions and in the window size used to calculate L_0. However, this time is far below the one spent in [11], which has used around 15 iterations. In the Figure 15, we show the tridimensional reconstruction performed using the resulting map shown in Figure 11.

Other experiments were performed to validate our proposals. Figure 16 shows the stereo pair of an indoor scene (a room). We note several objects in the background, making it hard to represent the scene in details in the final dispar-

ity map. Due to the continuity hypothesis, several objects will agglutinate in a same disparity range. We used a morphological operator to decrease or avoid the effects caused by object edges and the several different textures noted in the scene. Figure 17 shows $L_0(l,c,d)$ and $L_n(l,c,d)$. As above, we used a window of 3×3 for L_0 and a local support of $5 \times 5 \times 3$, and only two iterations. The time for L_0 was 3.91 sec and final time was 8.82 sec, with a cut point of 0.9. In the same way, we augmented cut point while keeping window for L_0 in 3×3 and local support in $5 \times 5 \times 3$ (Figure 18) with lost in precision (note erroneous horizontal dark bar due to cut point used). We also decreased local support, as shown in Figure 19. This improves the time, but also generates false correspondences and false occlusions. Figure 20 shows a comparison with an ideal result (left image is for our method). In this case, We can see that the precision of both results is almost the same.

Figure 16: Pair of an indoor scene (385×289).

Figure 17: L_0 on left, L_2 on right.

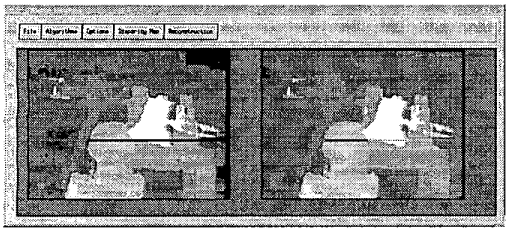

Figure 18: Increasing cut point.

Finally, we select the stereo pair of the Santa Helena volcano to show the algorithm applied to natural outdoor scenes. We note a texture that is a little different from the previous images texture, composed of repetitive stripes. Also, the projection distance is greater here. These aspects may introduce problems in the algorithm (for example, the window size for determination of L_0 shall be bigger). Figure 22 shows results that we got using a 5×5 window for determining L_0, in the left. We used a local support of $5 \times 5 \times 3$ and a cut point of 0.9 for the MCF in the DP convergence. The final result for this experiment is shown in the right of Figure 22. We note that false matchings and false occlusions appears. It was possible to predict these problems, in function of the parameters used. In [1], using the same images, Gonçalves and Oliveira describe experiments using windows varying from 12×12 to 32×32. They got best results for matching with images of 24×24. In this way, we next augmented these parameters. We used a window of sizes $9times9$ for determining L_0 and the local support was augmented to $9 \times 9 \times 5$. As a result, the function L_0 and the final results, both shown in Figure 23, were significantly improved. We can observe some occlusions correctly detected in the final images (due to shadows over the mountains).

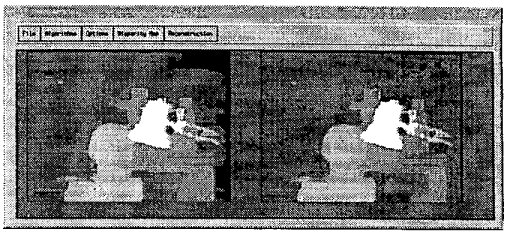

Figure 19: Decreasing local support to $3 \times 3 \times 3$

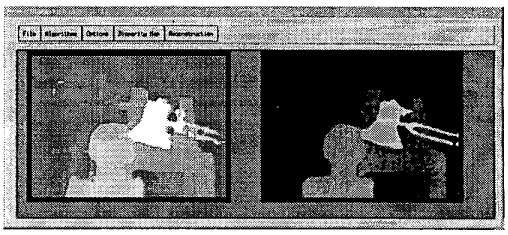

Figure 20: Comparing with an ideal result.

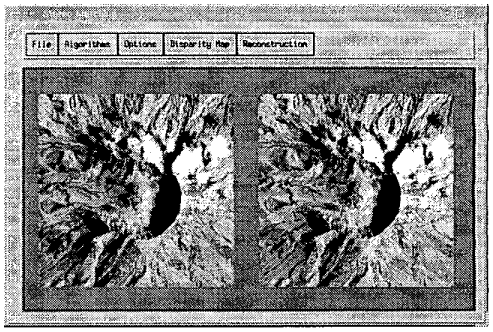

Figure 21: Santa Helena volcano (256×256).

7 Discussions, conclusions, and future work

The modifications introduced in Zitnick's algorithm makes a balancing between robustness, precision, and processing speed. We introduced basically two main improvements.

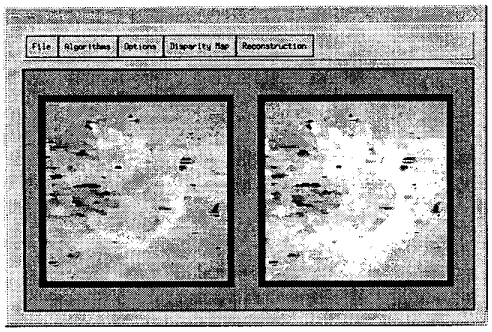

Figure 22: With local support of $5 \times 5 \times 3$.

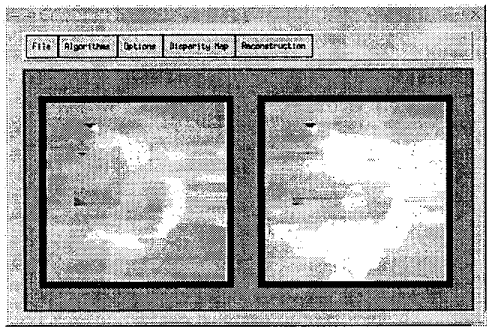

Figure 23: With local support of $9 \times 9 \times 3$.

First one is choosing a better initialization and normalization function. In the initial tests, we got several problems with this function as suggested in [9, 10, 11]. With our normalization function, we get at least 50a better manipulation of the other parameters. The second improvement is using DP that considers the uniqueness and continuity (smoothness) restrictions to accelerate the iterative process. With some lost in precision in the determination of occluded areas (suavization implies in lost of details), we get final results in a much faster time. Our final results close to the "ideal" one was obtained in three iterations, while Zitnick and Kanade got their best results in some 15 iterations [11]. Finally, a better choosing of the local support dimensions also accelerates the final map convergence. It is important to remark that the gain in time can justify some lost in precision. In other words, to detect occlusion in the same level, perhaps some few more iterations would be necessary. Besides, note that in some applications as robotics a faster approach (as the one provided here) is necessary. At this point, we can enumerate the main advantage and disadvantage of using DP. The main goal of this work was to improve time performance, so, for sure, the advantage is the substantial reduction in time. The disadvantage could be related to the lower level for detection of occlusions. However, that can be noted only in images with certain amount of noise or else if we use a small local support and/or a bigger cut point for the minimum cost function. Moreover, the proposed enhancement in Zitnick's algorithm yet maintain consistency in the determination of occluded zones and the use of DP proved to be helpful to get smoother results.

The first and exhaustive tests were dedicated to find the best parameters to be used by the algorithm to avoid primary errors. If we augment local support, execution time also augments. We note that the size of the local support is a function of the amount of texture (discriminability) in the images. As another good result, we reduced the amount of memory due to best exploring the Marr and poggio restrictions. In this way, we can finally state that our algorithm produce results very close to the one in [9, 10, 11] and that the best improvement was the gain in time by using DP.

To increase efficiency in the processing time, the algorithm can be adapted to fit in a distributed or parallel scheme. The use of distributed processing can yet allow it to be applied to sequences of stereo images, to be used in robotics applications. Another improvement that can be tried is to enhance the robustness of the DP technique. We used a simple approach here. Other approaches that consider discarding false matchings could be tried. This could improve detection of occlusions.

References

[1] L. M. G. Gonçalves and A. A. F. Oliveira. Pipeline stereo matching in binary images. *XI International Conference on Computer Graphics and Image Processing (SIBGRAPI'98)*, pages 426–433, October 1998.

[2] W. E. L. Grimson. Computacional experiments with a feature based stereo algorithm. *IEEE Transactions on Patten Analysis and Machine Intelligence*, 7(1):17–34, 1994.

[3] B. Julesz. *Foundations of Cyclopean Perception*. University of Chicago Press, Chicago, IL, USA, 1971.

[4] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *Transactions on Pattern Analysis and Machine Intelligence*, 16:920–932, 1994.

[5] D. Marr and T. Poggio. Cooperative computation of stereo disparity. In *Science*, volume 194, pages 209–236, 1976.

[6] K. Nishihara. Minimal meaningful measurements tools. Technical report, Teleos Research, 1991.

[7] K. Nishihara, H. J. Thomas, and E. Huber. Real-time tracking of people using stereo and motion. Ai lab technical report, Massachusetts Institute of Technology, 1984.

[8] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52(2):93–99, 1985.

[9] C. L. Zitnick and T. Kanade. A volumetric iterative approach to stereo matching and occlusion detection. *CMU Techincal Report CMU-RI-TR-98-30*, 1998.

[10] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *CMU Techincal Report CMU-RI-TR-99-35*, 1999.

[11] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000.