

A Volumetric Stereo Matching Method: Application to Image-Based Modeling

Qian Chen and Gérard Medioni*

University of Southern California, Los Angeles, CA 90089-0273
{chenqian, medioni}@iris.usc.edu

Abstract

We formulate stereo matching as an extremal surface extraction problem. This is made possible by embedding the disparity surface inside a volume where the surface is composed of voxels with locally maximal similarity values. This formulation naturally implements the coherence principle, and allows us to incorporate most known global constraints. Time efficiency is achieved by executing the algorithm in a coarse-to-fine fashion, and only populating the full volume at the coarsest level. To make the system more practical, we present a rectification algorithm based on the fundamental matrix, avoiding full camera calibration. We present results on standard stereo pairs, and on our own data set. The results are qualitatively evaluated in terms of both the generated disparity maps and the 3-D models.

1. Introduction

Shape-From-Stereo is one of the widely studied topics in computer vision. The literature is extensive. Assuming a fully calibrated stereo rig, early work concentrated on matching [4]. Shape inference was considered straightforward using the triangulation procedure. While uncalibrated cameras are used in our work, in this paper we confine ourselves to the matching problem, knowing that effective self-calibration methods exist [11,14].

Several issues need to be addressed when matching two stereo images, namely, what the local matching primitives are (pixels, edges or areas); what the global constraints are; and how to handle some special problems such as holes and occluding contours. Although it is known from neurophysiological observations that binocular cortical cells respond to oriented edges, we cannot simply draw the conclusion that they are the matching primitives used in human stereopsis. In fact, this question remains open. On the other hand, as pointed out in [4], the major distinguishing factor among different stereo algorithms is the way global constraints are managed, because local similarity measurements almost always produce ambiguous matches. This is true even

when the epipolar constraint is imposed, as it only reduces a match to a line in the other image.

Frequently used global constraints include: the gradient limit constraint [1], the ordering constraint [16], uniqueness and the coherence principle [12]. Among them, the one that is most general is Prazdny's coherence principle which, as suggested by the author, often manifests itself into the rule of continuity and smoothness as a result of the cohesiveness of matter [8] of opaque surfaces. The algorithm proposed by Prazdny proceeds in two stages. First, all potential disparities for each point are found. Next, the disparities are allowed to influence each other. After collecting all information from the neighbors, the disparity with the largest support is chosen as the most likely disparity at a point. Unfortunately, the core of his implementation—a scalar-valued similarity function—seems to be too unsophisticated to carry out the power of the principle itself.

A more capable solution is provided by Lee *et. al.* [7] where each data site is represented by a tensor (or ellipsoid) that captures location, surface normal and uncertainty. During *tensor voting*, consistent sites enhance one another, increasing the confidence on the normal estimation, while inconsistent sites interfere with one another, increasing the uncertainty. One disadvantage of Lee's method is that the voting process is time-consuming. The reason comes from the fact that the convolution kernel is 3-dimensional and that it has to be aligned with the normal at each data site by applying a 4×4 transformation.

In implementing the coherence principle, it is natural to treat the disparity d as a function of the pixel coordinates (u,v) in one of the images (e.g. the left one). Then $d(u,v)$ defines the so-called disparity surface which is a collineation of the visible portion of the objects in the scene [13]. The disparity surface is continuous, except at occluding boundaries; and smooth, except along ridge contours. By enforcing the continuity and/or smoothness conditions, one is able to compute the surface as in the feature-based approaches [10,15]. In particular, the latter approach parameterizes $d(u,v)$ as a function with Gaussian basis. The result, however, looks overly smooth. Additionally, the figural continuity can be used to handle depth or normal discontinuities [6,9]. A second merit of

* This research has been funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency.

[6] is that it is more resistant to outliers because they are first removed by the fitting of planar patches in the Hough space over all possible local matches. On the downside, this step becomes the bottleneck of the algorithm. In [3], dynamic programming is used to compute the disparity surface line-by-line. The main consideration here, is the computational complexity. The disadvantage is that intra-scanline coherence is ignored, leading to artifacts in the resulting disparity map.

Most existing surface oriented approaches assume the disparity surface is unknown and try to approximate it one way or the other. In this paper, we argue that the disparity surface is by default embedded inside a 3-D volume, and when the volume is populated with correlation values, it is simply composed of locally maximal voxels. Consequently, the stereo matching problem is converted into an extremal surface extraction problem, which can be solved efficiently using volume-rendering techniques, resulting in a dense disparity map. This formulation naturally implements the coherence principle, and allows using most known global constraints. Compared to the scanline-based approaches, spatial coherence is better utilized. Compared to regularization approaches, ours captures the local shape variation more accurately. Unlike those methods which model the disparity surface as an algebraic function, the disparity surface is discrete in our case.

We begin the presentation with an image rectification algorithm that does not require camera calibration (section 2). We then describe the matching method (section 3). Experimental results are presented afterwards (section 4). Finally, conclusions and future research directions are offered (section 5).

2. Image Rectification

In general, the two cameras used to take the stereo images are not parallel. The task of rectification is to numerically align the two image planes so that they are coplanar and their scanlines are horizontally coincident. Traditionally, this step requires fully calibrated cameras. Recently, Hartley et al [5] gave an algorithm by finding a special factorization of the fundamental matrix. Here, we present a more intuitive algorithm that is also based on the fundamental matrix. It works well for two close-by images, which is the case of stereo.

In Figure 1, $l1$, $r1$ and $l2$, $r2$ are two pairs of corresponding epipolar lines. $v1$ is the average of the y coordinates of the four end points of $l1$ and $r1$, and $v2$ is that of $l2$ and $r2$. The rectification transformation, one for each image, maps the trapeze (dark) to the rectangle (grey). Next, we briefly explain why this algorithm works.

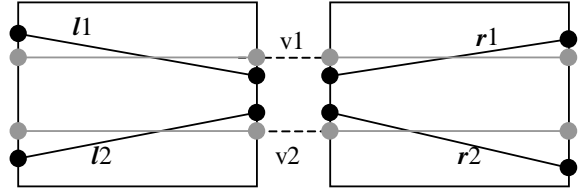
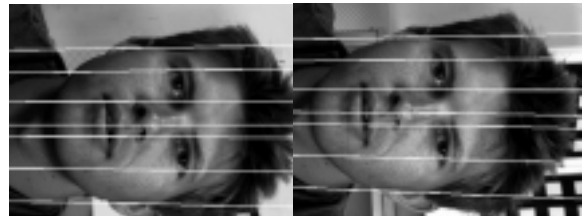


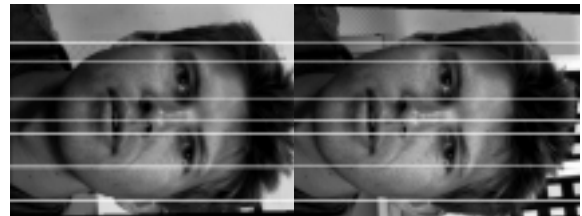
Figure 1. Illustration of the rectification algorithm

First, we know that the epipolar lines of an image form a pencil-of-lines whose intersection is the epipole; and the pencil-of-lines are the projections of the 3-D pencil-of-planes whose intersection is the line connecting the two projection centers. Second, by designating the special line that connects the epipole and the image as one of the projective bases (which is a line-at-infinity), any line in the pencil is a linear combination of the two remaining bases. The coefficient is indeed a cross-ratio, thus a projective invariant. Third, after the rectification, all three bases are aligned—the special one is mapped to the X -axis, the corresponding epipolar lines (or scanlines) are necessarily aligned, otherwise they would have different coefficients.

The result of rectification is depicted in Figure 2 where the epipolar lines are superimposed. Clearly the goal has been achieved, that is, the scanlines are parallel in each image, and the corresponding ones are collinear across the images. The initial correspondences needed for the fundamental matrix are automatically computed using the software provided by Zhang [17].



(a) before rectification



(b) after rectification

Figure 2. Example image rectification

3. The Matching Algorithm

3.1 The “u-v-d volume”

We use normalized cross-correlation over a window as the similarity measurement of two potential matches:

$$\Phi(u, v, d) = \frac{\text{Cov}(W_l(u, v), W_r(u + d, v))}{\text{Std}(W_l(u, v)) \cdot \text{Std}(W_r(u + d, v))}$$

where W_l and W_r are the intensity vectors of the left and right windows of size ω centered at (u, v) and $(u+d, v)$ respectively, d is the disparity, “Cov” stands for covariance and “Std” for standard deviation. The width and height of the (left) image together with the range of d form the u - v - d volume. The range of Φ is $[0, 1]$. When Φ is close to 1, the two pixels are well correlated, hence have high probability of being a match. When Φ is close to zero, that probability is low. In implementation, a threshold needs to be set. We discuss how to choose its value in the next section.

Shown in Figure 3 is the left image of the Renault auto part stereo pair, and a slice (whose position is marked) in the corresponding u - v - d volume where lighter pixels corresponding to higher correlation values. The cross-section of the disparity surface is clearly observable.

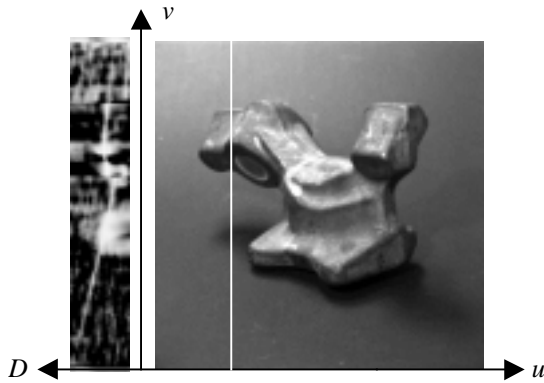


Figure 3. A slice in u - v - d volume of *Renault part*

3.2 Disparity surface extraction

The fact that Φ is a local maximum when (u, v, d) is a correct match means that the disparity surface is composed of voxels with peak correlation values. Matching two images is therefore equivalent to extracting the maximal surface from the volume. Since the u - v - d volume is very noisy, simply applying the Marching Cubes algorithm would easily fall into the trap of local maxima. We thus implemented a propagation type of algorithm. In addition, we make use of the disparity gradient limit which states that $|\Delta d|/|\Delta u| < 1$.

3.2.1 Algorithm description

The output from our matching algorithm is a disparity map which corresponds to the voxels that comprise the disparity surface. This is where it differentiates itself from volume rendering, or other matching methods that model the disparity surface as a continuous function. The algorithm undergoes two steps.

Step 1. Seed voxel selection

A voxel (u, v, d) is a seed voxel if, (i) it is unique – meaning for the pixel (u, v) , there is only one local maximum at d along the scanline v , and (ii) $\Phi(u, v, d)$ is greater than a threshold $t1$. A seed necessarily resides on the disparity surface. Otherwise, the true surface point (u, v, d') for which $d' \neq d$ would be a second local maximum. To find seeds, the image is divided into a number of buckets. Inside each bucket, pixels are checked randomly until either one seed is found, or all pixels have been searched without success. During the search, the voxel values are cached to save computation time for the next step. The value of $t1$ determines the confidence of the seed points and is set close to 1. In our experiments, we start from 0.995 trying to find at least 10 seeds. If too few seeds are found, its value is decreased. In all the examples presented later, we have found the range of $t1$ to be between 0.993 and 0.996.

Step 2. Surface tracing

```

Initialize  $Q$  with the seed voxels;
While (not empty  $Q$ )
{
  Pop the head of  $Q$ ;
  Let it be  $(u, v, d)$ ;
  for each 4-neighbor of  $(u, v)$ 
  {
    call it  $(u', v')$ ;
    choose among  $(u', v', d)$ ,  $(u', v', d-1)$ ,  $(u', v', d+1)$ 
    the one with the max correlation value,
    call it  $(u', v', d')$ ;
    if  $(u', v')$  already has a disparity  $d''$ 
      disparity $(u', v') = \Phi(u', v', d') > \Phi(u', v', d'')$ ?  $d' : d''$ ;
    else if  $\Phi(u', v', d') > t2$ 
    {
      disparity $(u', v') = d'$ ;
      push  $(u', v', d')$  to the end of  $Q$ ;
    }
  }
}

```

Figure 4. Pseudo code of the tracing algorithm

Simultaneously from all seed voxels, the disparity surface is traced by following the local maximal voxels whose correlation values are greater than a second threshold $t2$. The $|\Delta d|/|\Delta u| < 1$ constraint determines that when moving to a neighboring voxel, only those at d , $d-1$, $d+1$ need to be checked. Initially, we store the seed voxels in a FIFO queue. After tracing starts, we pop the head of the queue every time, and check the 4-neighbors of the corresponding pixel (border pixels need special treatment). When two surface fronts meet, the one with the greater correlation value prevails. If any new voxels are generated, they are pushed to the end of the queue. This process continues until the queue becomes empty. To enforce

smoothness, we assign (u',v',d) higher priority than $(u',v',d-1)$ and $(u',v',d+1)$. To obtain sub-pixel accuracy, a quadratic function is fitted at the newly generated voxel. t_2 determines the probability that the current voxel is on the same surface that is being traced. It turns out that the value of t_2 is not critical. In all the later examples, only one value (0.6) is used. The pseudo code listing (Figure 4) summarizes the algorithm.

3.2.2 Time complexity

The complexity of the seed selection part is bounded by $O(WHD\omega)$ where W and H are respectively the width and height of the image, D is the range of the disparity, and ω is the size of the correlation window. That of the tracing part is bounded by $O(WH\omega)$. Since some voxels have already been computed during the first stage, this limit is never reached. Actual running time on examples are given later.

3.2.3 Extraction result on the Renault auto part

Figure 5(a) shows the u - v - d volume (truncated to only include the part itself) after non-maxima suppression, which is still quite noisy. This indicates that simply using local similarity measurement for correspondence search could be disastrous. (b) is the volume after tracing. Now, the disparity surface appears clearly. (c) and (d) demonstrate the change of the cross-section shown earlier in Figure 3. In this example, $W=H=256$, $\omega=9$, and the disparity range is $[-30,40]$. Running on a SGI O2 with default values for t_1 and t_2 , step 1 takes 320 seconds which results in 880 seeds, step 2 takes 20 seconds.

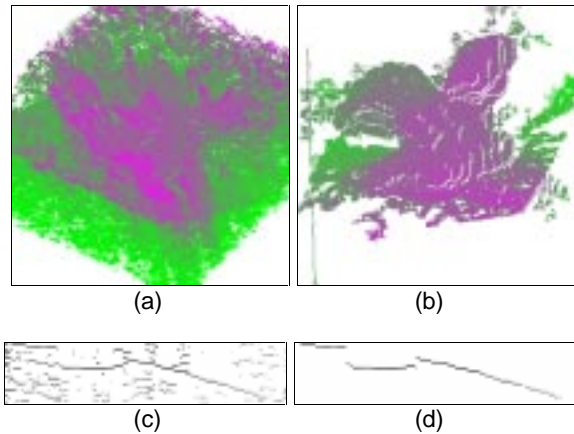


Figure 5. Matching of the "Renault auto part"

3.3 Improving the time efficiency

Obviously, the bottleneck of the previous extraction algorithm lies in the seed selection part. To improve the time efficiency, we modify the algorithm so that it proceeds in a multiresolution fashion: only at the coarsest level is the full volume computed; at all subsequent levels, the seeds are inherited from the previous one. To guarantee

the existence of seeds at the coarsest level, we replace the uniqueness condition (in step 1 of 3.2.1) by a winner-take-all strategy, that is, at each (u,v) , we compute all voxels (u,v,d) where $d \in [-W_0/2, W_0/2]$ and choose the one that has the maximum correlation value. Under this relaxed condition, some seeds may represent incorrect matches. To deal with this, we randomly assign the seeds to five different layers. As a result, five disparity maps are generated at the end of tracing. This gives us a chance to identify the wrong match for each pixel as an outlier. If no agreement can be reached, we leave that point as unmatched. At each level, extraction is performed for both left and right images. Cross-checking is then conducted. Those pixels whose left and right disparities differ more than one pixel are eliminated and recorded as unmatched. At the finest level, small holes are filled by erosion from their borders. Figure 6 shows the disparity maps (of left and right images respectively) resulting from the improved algorithm. The execution time is reduced to 124 seconds.

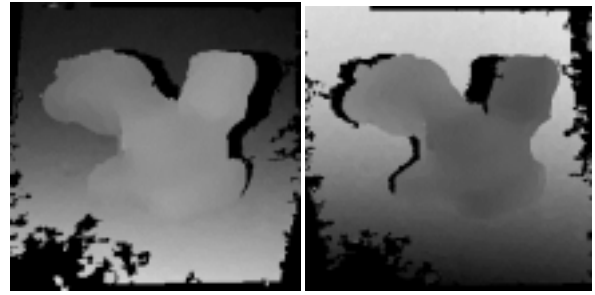


Figure 6. Disparity maps of the "Renault auto part"

Assume the reduction rate is 4 and the size of the correlation window is constant over all resolutions, the time complexity is about $O(6WH\omega)$. Another merit of the multi-resolution version is that there is no need to prescribe a value for D .

4. Results

In this section, we demonstrate the proposed matching algorithm by showing the computed disparity maps as well as the 3-D reconstruction on several examples. There are two reasons that motivate us to use 3-D models. First, we are aiming at building an Image-Based Modeling tool for multimedia applications for which disparity maps are not sufficient. Second, we believe 3-D models can be better judged for quality. However, it is not our intention in this paper to describe fully the algorithm that converts a disparity map into Euclidean coordinates. In a very high level, we first compute a projective reconstruction from the correspondence information. Then the projective reconstruction is brought to an Euclidean one by enforcing some orthogonal constraints that come with the pinhole

camera models. Related approaches are described in [2,11,14].

Example 1 – Renault auto part

From one of the disparity maps displayed in Figure 6, we interactively cull the portion corresponding to the part itself, then perform the reconstruction. Shown in Figure 7 are two texture-mapped views of the result.



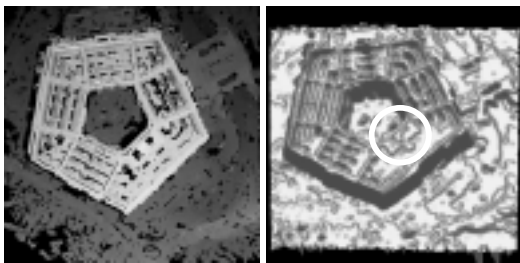
Figure 7. Renault auto part

Example 2 – Pentagon

Figure 8 (a) shows the input pair, (b) is the disparity map and (c) is a shaded view of the 3-D reconstruction which is a direct reflection of the disparity surface since the projection is almost orthographic. Notice that details such as the concourse, the freeway, and the small T shape roof (inside the white circle) have all been recovered.



(a)



(b)

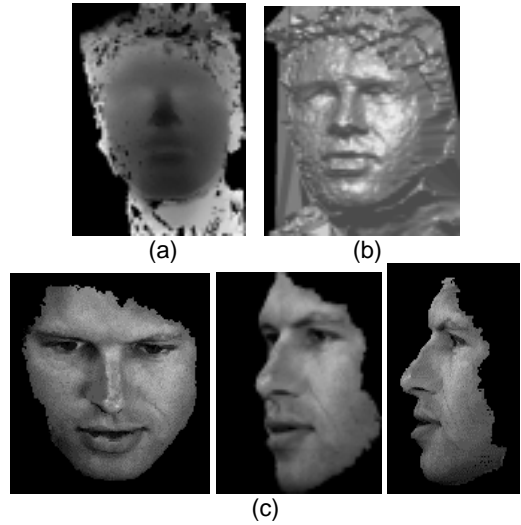
(c)

Figure 8. Pentagon

Example 3 – human faces

Here, we apply our approach to the reconstruction of human faces. The first example is Hervé’s face which we have seen in Figure 2. Figure 9(a) shows the resulting disparity map. (b) is a shaded view of the 3-D reconstruction. (c) contains several texture-mapped views of the frontal face. Figure 10 is another face example

where the person is smiling. Both shaded and textured views are shown. It is clear that major face features such as the eyebrows, nose, mouth have been faithfully recovered.



(a)

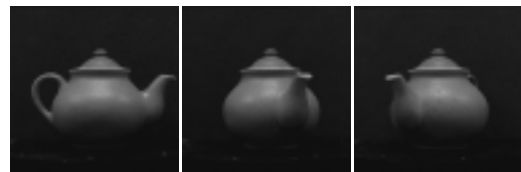
(b)

(c)

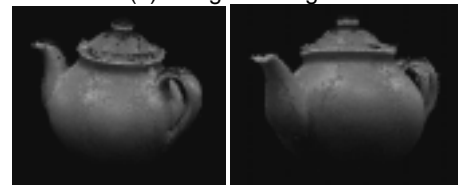
Figure 9. Hervé’s face



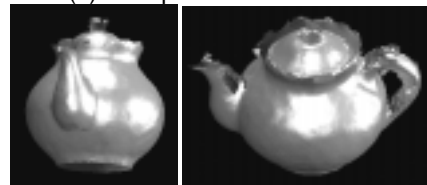
Figure 10. Reconstruction of “Doug’s face”



(a) Original images



(b) 3-D point reconstruction



(c) Surface reconstruction

Figure 11. Original and reconstruction of a teapot

Example 4 – a complete teapot

We put a teapot on a turntable and took six stereo images of it. Three of them are shown in Figure 11(a). In order to apply the known rotation to each partial reconstruction, it was necessary to estimate the transformation between the default world coordinate system—that of the first camera—and the rotary table’s coordinate system. This was achieved by placing a cube with one of its corners at the center of the platform. Figure 11(b) depicts the reconstructed teapot after fusing all points. Each point was assigned the intensity of the source pixel from the appropriate image. From these points, a surface (Figure 11(c)) was constructed. Notice that the overall shape has been captured very well except the top areas which cannot be seen by the cameras.

Execution time

Summarized in Table 1 is the running time (in seconds, on a SGI O2 workstation) of all examples. The code was not optimized. For example, the queue was implemented as a linked list.

Table 1. Summary of execution time

Example	# Images	Size	Time
Renault part	2	256×256	124s
Pentagon	2	512×512	726s
Hérve’s face	2	512×384	469s
Doug’s face	2	512×512	732s
Teapot	12	512×512	2501s

5. Conclusion and Future Work

We have presented a volumetric algorithm for stereo image matching. The key difference from other related approaches is that we assume the disparity surface is already embedded inside the correlation volume. The matching algorithm simply extracts it as an extremal surface. This embedding allows us to make use of most known global constraints in a natural way, especially the coherence principle, while also captures the local shape variation. The algorithm uses a small number of parameters, namely, t_1 for seed selection, t_2 for tracing, and ω for the size of the correlation window. As an integral part of the system, we also present a new algorithm for image rectification that does not require full camera calibration.

In the current implementation, edges are not considered. Thus ridge contours tend to be rounded. However, it is not very difficult to incorporate the edge information into our framework. For example, we can first detect edges, and stop surface propagation when an edgel is met. Alternatively, we can explicitly model the

occluding and ridge contours in a separate step, as what has been done in [6].

References

1. P. Burt and B. Julesz, "A disparity gradient limit for binocular fusion", *Perception*, 9, 671-682, 1980.
2. Q. Chen and G. Medioni, "A Semi-automatic System to Infer Complex Shapes from Photographs", to appear in *IEEE Multimedia Systems'99*, Florence, Italy, June, 1999.
3. I. Cox, S. Hingorani, S. Rao, B. Maggs, "A Maximum-Likelihood Stereo Algorithm", *CVIU*, 63(3), 542-567, 1996.
4. U. Dhond and J. Aggarwal, "Structure from stereo - a review", *IEEE T. Systems, Man and Cybernetics*, 19(6), 1489-1510, 1989.
5. R. Hartley and R. Gupta, "Computing Matched-epipolar Projections", *CVPR'93*, 549-555, 1993.
6. W. Hoff and N. Ahuja, "Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection", *PAMI*, 11(2), 121-136, 1989.
7. M.-S. Lee and G. Medioni, "Inferring Segmented Surface Description from Stereo Data", *CVPR'98*, 347-352, 1998.
8. D. Marr and T. Poggio, "A computational theory of human stereo vision", *Proc. Royal Soc. London*, B204, 301-328, 1979.
9. Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-scanline Search Using Dynamic Programming", *PAMI*, 7(2), 139-154, 1985.
10. S. I. Olsen, "Stereo Correspondence by Surface Reconstruction", *PAMI*, 12(3), 309-314, 1990.
11. M. Pollefeys, R. Koch and L. V. Gool, "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters", *ICCV'98*, 90-95, 1998.
12. K. Prazdny, "Detection of Binocular Disparities", *Biological Cybernetics*, 52, 93-99, 1985.
13. R. Szeliski and P. Golland, "Stereo matching With Transparency and matting", *ICCV'98*, 517-524, 1998.
14. B. Triggs, "Autocalibration and the Absolute Quadric", *CVPR'97*, 609-614, 1997.
15. G.-Q. Wei, W. Brauer and G. Hirzinger, "Intensity- and Gradient-Based Stereo Matching Using Hierarchical Gaussian Basis Functions", *PAMI*, 20(11), 1143-1160, 1998.
16. A. L. Yuille and T. Poggio, "A generalized ordering constraint for stereo correspondence", *A.I. Memo 777*, AI Lab, MIT, 1984.
17. Z. Zhang, R. Deriche, Q.-T. Luong and O. Faugeras, "A Rotust Approach to Image Matching: Recovery of the Epipolar Geometry", *Proc. Int'l Symposium of Young Investigators on Information-Computer-Control*, 7-28, Beijing, China, 1994.