3D Structure Recovery From Stereo Using Synchronous Optimization Processes

Tarkan Aydin
Department Of
Computer and Mathematics
Bahcesehir University
Besiktas, Istanbul 34349 Turkey
tarkan.aydin@bahcesehir.edu.tr

Yusuf Sinan Akgul
GIT Vision Lab
Department Of Computer Engineering
Gebze Institute Of Technology
Cayirova, Gebze, Kocaeli 41400 Turkey
akgul@bilmuh.gyte.edu.tr

Abstract

This paper presents a novel system that uses two synchronous optimization processes to recover 3D structure from rectified stereo image pairs. The synchronization of the processes are done by energy terms that inform the optimization processes about the recovered positions of each other. This information is used to direct the optimizations towards a better direction. The system is initialization insensitive and it is very robust against local minima. We performed experiments on real and synthetic images with ground truth that showed the effectiveness and the robustness of our system. We also compared our system to other systems for further validation.

1 Introduction

Although the estimation of the 3D structure using stereo is one of the oldest techniques of Computer Vision, the problem is still being researched intensely by many groups. The simplicity and availability of the image acquisition hardware, very strong epipolar geometric constraints, naturally inspiring systems such as human vision, and the wide range of applicability of these systems are a few reasons for the popularity of stereo.

Establishment of the correspondence between stereo image pairs is considered as the first and most important problem of classical stereo analysis[8, 2]. More current techniques take the path of formalizing the stereo problem as the global solution of estimating the 3D structure directly from the images, e.g., [10, 9, 14]. The formulations are usually written as one global energy functional that needs to be optimized to produce the desired 3D surface. The optimization of the functionals are generally NP-Hard for most of the cases in stereo[3]. As a result, some researchers simplified the functionals so that a globally optimal solution is possible, e.g., using dynamic programming [6][12]. However, for most cases, it is not possible to simplify the stereo analysis model. Therefore, using an approximate optimization method became more popular. These methods include[13] stereo by simulated annealing, graph cuts, gradient descent, genetic algorithms, etc. Although some of these methods produce very good results, optimality is still not satisfied and getting closer to optimal results is always desirable.

This paper describes a system that uses two separate optimization processes for the recovery of 3D surfaces. The optimization processes are based on gradient descent heuris-

BMVC 2006 doi:10.5244/C.20.120

tics, which do not guarantee optimality. However, due to the interaction between the optimization processes, the overall result of our system is always better than the results achievable by a single optimization process. The presented idea is applicable to other heuristic optimization methods that use a global approach.

Our system is not the first one to use two optimization processes in synchronization. Akgul and Kambhamettu [1] used such a system with two deformable meshes. Our system introduces a more methodological way of synchronizing the optimization processes by using an automatic way of changing the amount influence between the processes. We also use a more effective regularization term for the structure smoothness. The resulting system is more efficient and less sensitive to local minima. Our system requires also minimal information from the user which makes it more robust against the differences between images.

The rest of this paper is organized as follows. Section 2 defines the energy functional and its subterms. Section 3 defines the details of the optimization process. Section 4 describes the system validation and experiments and we conclude our paper with Section 5.

2 The Energy Functional

The 3D surface to be recovered should be locally smooth and a given 3D reconstructed point should be projected onto similar image regions on the left and right stereo images. We express these features in the form of global energy functionals following the classical regularization method [16]. These energy functionals will produce two discrete valued disparity functions $d_1(i,j)$ and $d_2(i,j)$ that will show the results of the two separate but dependent optimizations. These disparity functions will assign a disparity value for each element L_{ij} in the left image of the stereo pair. We write the energy functionals as

$$E(d_1) = \sum_i \sum_j E_{Data}(L_{ij}, R_{ij-d_1(i,j)}) + \lambda_1 E_{Smth}(d_1(i,j)) + \lambda_2 E_{Tnsn}(d_1(i,j), d_2(i,j)) \quad (1)$$

$$E(d_2) = \sum_i \sum_j E_{Data}(L_{ij}, R_{ij-d_2(i,j)}) + \lambda_1 E_{Smth}(d_2(i,j)) + \lambda_2 E_{Tnsn}(d_2(i,j), d_1(i,j)) \quad (2)$$

The data term $E_{Data}(L_{ij}, R_{ij-d_1(i,j)})$ is for satisfying the image similarity requirement.

$$E_{Data}(L_{ij}, R_{ij-d_1(i,j)}) = 1 - Corr(L_{ij}, R_{ij-d_1(i,j)}) \quad (3)$$

Assuming the images are row rectified, data term uses the popular normalized cross correlation, $Corr$, values between the left(L_{ij}) and right($R_{ij+d_1(i,j)}$) image regions. Note that, for pixel L_{ij} of the left image, the disparity function d_1 chooses the pixel $R_{ij-d_1(i,j)}$ from the right image on the same row. Since the normalized cross correlation produces values in the range [-1, +1], the E_{Data} term would be close zero when the two image regions are very similar. If the images are very dissimilar, it gets close to 2. Our data energy is robust against any brightness differences between the left and right images because it is normalized.

The smoothness term of the energy functional makes the resulting disparity functions smooth both in x and y dimensions. For the smoothness metric $E_{Smth}(d_1(i,j))$, we use the error of plane fit around the disparity image point $d_1(i,j)$. This metric becomes zero for locally planar regions of the disparity image. Our smoothness term does not allow surface

discontinuity because it extends the smoothness everywhere in the images. However, for images with large smooth regions, our system still produces satisfactory results.

It is not very difficult to make our system produce discontinuous surfaces natively. There are a number of popular alternatives such as Potts model, which was first used by Geman et al.[5] for vision.

The tension energy, E_{Tnsn}, is for the synchronization of the two optimization processes. The main function of the tension term is to make the disparity values of two functions d_1 and d_2 get close to each other by pushing the optimization process with the worse data term towards the other process.

$$E_{Tnsn}(d_1(i,j),d_2(i,j)) = |d_1(i,j) - d_2(i,j)| E_{Data}(L_{ij},R_{ij-d_1(i,j)}) \left(\lambda_3 + 2 - E_{Data}(L_{ij},R_{ij-d_2(i,j)})\right), \tag{4}$$

where λ_3 is the constant tension that pushes the disparity functions together when both data terms are very high. Note that the tension term is heavily dependent on the data energy term. If the data term of the optimization is close to zero, then the optimization is not affected from this term. If the optimization has a high data energy and the other optimization has the lower data energy, then the tension term will push the optimization towards the disparity values of the other optimization.

3 Synchronous Energy Optimization

The final disparity images of the optimizations of Equation 1 and 2 are the d_1 and d_2 disparity images that satisfy

$$\min E(d_1)$$
$$\min E(d_2)$$

If the energy functionals defined by Equation 1 and 2 are optimized independently by heuristic methods starting from different initial configurations, each would produce a different disparity map. However, if we optimize them in synchronization with the help of the tension term, they can be forced to find the same surface. The biggest advantage of such a mechanism is that the overall optimization would localize a much better 3D position than each of the optimizations can achieve. This advantage comes from the system feature that can compare the positions of two optimization processes and bias the optimization direction towards to better position.

Note that in order this idea to work, we need a basic optimization method. For the current system, we chose the gradient descent algorithm because of its simplicity. In addition, gradient descent is not a very powerful optimization method for stereo due to its sensitivity to initial configurations and local minima, so a solution using this optimization method would show the effectiveness of our system.

In classical gradient descent optimization algorithm, we search for a vector position that satisfies the optimal values of a functional by moving along the directions of the gradients of the functional. For our case, this vector is the elements of the disparity images d_1 and d_2. Since we have two optimization processes, there will be two search vectors $\vec{v_1}$ and $\vec{v_2}$.

$$\begin{aligned} \vec{v_1} &= [\ldots, d_1(i,j), \ldots]^T \\ \vec{v_2} &= [\ldots, d_2(i,j), \ldots]^T \end{aligned}$$

We also define two objective functions $E_1(\vec{v_1})$ and $E_2(\vec{v_2})$ that can take vectors and produce the energy value for the given vector as defined by Equations 1 and 2. We then define the gradient direction for these vector functions as

$$\nabla E_1(\vec{v_1}) = \left[\frac{\partial E_1(\vec{v_1})}{\partial \vec{v_1}(1)}, \frac{\partial E_1(\vec{v_1})}{\partial \vec{v_1}(2)}, \ldots \right]^T, \tag{5}$$

where $\vec{v_1}(1)$ is the first element of the vector $\vec{v_1}$. Note that the number of elements of the vector $\vec{v_1}$ is the same as the number of pixels in the left stereo image.

A similar direction is also defined for E_2. The gradient directions show the direction where the value of the function E decreases most. The gradient descent algorithm calculates these directions repeatedly and takes steps on these directions. This process keeps changing the disparity vectors $\vec{v_1}$ and $\vec{v_2}$ until the gradient descent algorithm cannot lower the overall energies.

Note that classical gradient descent algorithm has serious problems with local minima and initial position of the vectors $\vec{v_1}$ and $\vec{v_2}$. These problems are not specific to gradient descent algorithm and we show that our synchronous optimization method can handle these problems.

The following are the steps of synchronous optimization

1. Set the initial values of the vector $\vec{v_1}$ to *minimum* possible disparity values. This will make the disparity image d_1 a constant image.

2. Set the initial values of the vector $\vec{v_2}$ to *maximum* possible disparity values. Similar to d_1, this will make the disparity image d_2 a constant image.

3. Calculate the values of $E_1(\vec{v_1})$ and $E_2(\vec{v_2})$. Note that the values of E_1 depends on both disparity images d_1 and d_2 due to the tension term. The same is true for E_2.

4. Calculate the gradient directions $\nabla E_1(\vec{v_1})$ and $\nabla E_2(\vec{v_2})$ using Equation 5. Move the vectors $\vec{v_1}$ and $\vec{v_2}$ on these directions with a step size dependent on the gradient magnitudes.

5. If the current d_1 and d_2 images produced by $\vec{v_1}$ and $\vec{v_2}$ are not the same, then continue with the step 3 as the next iteration.

The above process is not initialization sensitive because the first steps (initialization steps) are always independent of the input images. The above procedure also eliminates a considerable amount of problems due to local minima because when the gradient descent search is stuck due to local minima, it is always possible to compare the position with the other process to decide if it is a local minima or not.

The biggest difference between the above method and the method of [1] is that we do not turn the tension term manually on and off. The amount of tension is dynamic

between the processes, which removes a sensitive parameter from the system and makes it more robust against variations between images. Another advantage of the dynamic tension term is the appropriate application of the tension amount depending on the image sections. The amount of tension is very small if the image areas are very textured and produce good correlation values. On the other hand, if the image sections are textureless, then the tension will be higher.

4 Experiments and Validation

In order to verify the system performance, we performed experiments on standard images from the literature with and without ground truth. We tested the system on many different images. We will show images where the system works normally as well as images where our system partially fails.

All the experiments were performed using the same set of system parameters. We chose a 9x9 correlation window for the data energy term for all experiments. Unless otherwise stated, we used stereo images from the Middlebury image base[13] for these experiments.

The first experiment is on a baseball stereo image pair[7] without ground truth. Despite the depth discontinuities between the baseball and the background, our algorithm produces visually correct results. Figure 1 shows the original images and the disparity map produced by our system. The continuous surfaces are recovered nicely but there are a few visual problems at image positions with occlusions.

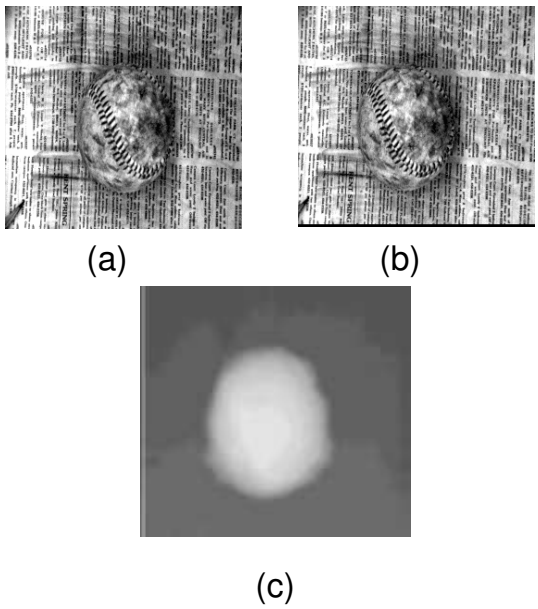

Figure 1: (a) Left baseball image (b) Right baseball image (c) The disparity map produced by our system.

It is interesting to see the two optimization processes in action, which is shown in Figure 2. Please see the supplementary material movie *baseball* for an animation of this figure. As the figure shows, one process starts from a constant disparity map d_1 with very low values, hence its initial map is a dark image. The second process starts from a constant disparity map d_2 with high values, hence its initial map is a bright image (Figure 2-a). When the optimizations start, the first process disparity values get higher and the values for the second one get lower(Figure 2 b-g). The process continues until the both disparity images become the same (Figure 2-h). Note that if one of the processes finds the correct disparity values, that position is kept and the other process disparity values are pulled to the found value. Note also that for regions where there is not much texture, the smoothness drives the optimizations.

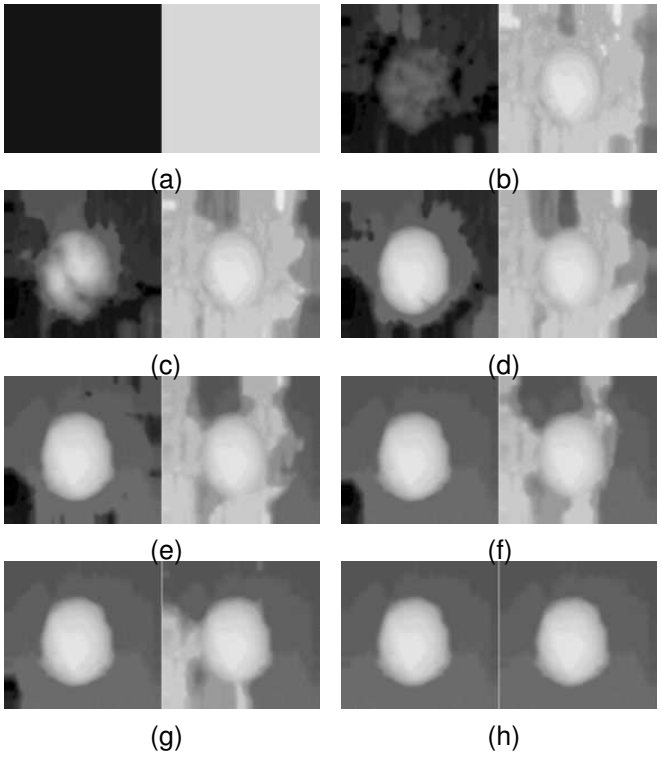

Figure 2: (a) The initial disparity maps of two processes. The dark image is for the first process with the minimum possible disparity values. The bright image is for the second process with the maximum possible disparity values. (b-g) The disparity maps of each process while the synchronized optimizations continue. (h) The final disparity maps of both processes. Note that both disparity maps are the same. Please see the supplementary material movie *baseball* for an animation of this figure.

The second example is the synthetic corridor image[4] with ground truth. This image has a number of depth discontinuities and large textureless sections. Figure 3 shows the left image, the ground truth, and the recovered disparity from our system. Visual

inspection of this image indicates that other than the immediate regions around the depth discontinuities, the disparities are correct.

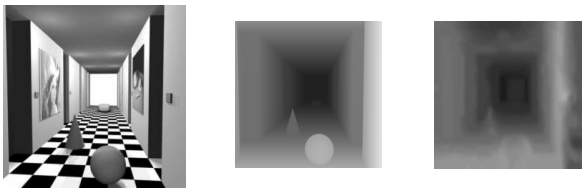

Figure 3: (a) The original left image of corridor. (b) The ground truth disparity map. (c) The disparity map recovered by our system.

We compared our disparity results from the corridor image with the ground truth disparity map. We also compared our results with a number of other stereo algorithm performances. Table 1 shows the comparison. The numbers for the other methods are by Li and Zucker[11], which were produced by comparison package of Scharstein and Szeliski[13]. The other algorithms are standard SSD, stereo for slanted surfaces (SSS)[11], graph cuts (GC)[3], and belief propagation (BPA and BPS)[15].

Disparity Error	SSD	SSS	GC	BPA	BPS	Our
RMS error	1.3	0.35	0.65	0.75	0.62	0.53
% error ± 1	14.0	3.4	7.4	10.1	5.4	7.2
% error ± 0.5	32.5	11.6	26.4	30.3	24.3	18.6

Table 1: Comparison of our algorithm with the ground truth and other algorithms on corridor image.

Note that the performance numbers are better than some of the other methods despite the existence of discontinuities. We are especially encouraged with the good RMS numbers because they show that even for problem areas, our algorithm produces values closer to the optimal.

In order to show the failure cases for our system, we chose an image (sawtooth image) with very strong and dominant depth discontinuities. Figure 4 shows the image, the ground truth, and the estimated disparity from our system.

Table 2 shows the comparison of our system on sawtooth image of Figure 4. Due to dominance of the depth discontinuities, our error percentage numbers are higher than other methods. However, our RMS numbers are still in the acceptable range. Please see the supplementary material movie *sawtooth* for an animation of this figure. This animation visually shows that the depth discontinuities are actually handled properly in the early phases of the optimization. However, when the smoothness term becomes dominant in the later phases of the optimization, the depth values around the discontinuities start getting worse.

Finally, figure 5 shows another example with smooth surfaces and discontinuities where our system performs favorably. Please see the supplementary material movie *venus*

Figure 4: (a) The original left image of sawtooh. (b) The ground truth disparity map. (c) The disparity map recovered by our system. Please see the supplementary material movie *sawtooth* for an animation of this figure.

Disparity Error	SSD	SSS	GC	BPA	BPS	Our
RMS error	1.65	1.30	1.42	1.67	1.45	1.59
% error +/-1	8.7	4.5	3.9	4.5	4.7	10.6

Table 2: Comparison of our algorithm with the ground truth and other algorithms on corridor image.

for an animation of this figure. The video shows the effect of smoothness after most of the disparities become the same.

Figure 5: (a) The original left image of venus. (b) The ground truth disparity map. (c) The disparity map recovered by our system. Please see the supplementary material movie *venus* for an animation of this figure.

5 Conclusions

We presented a novel system for the recovery of the 3D structures from stereo image pairs. The system defines an energy functional which is optimized by two synchronous optimization processes. The optimization processes always feed information to each other so that they know which search direction might produce better results. The final recovered 3D structure turns out to be much more accurate than what a single optimization process can achieve.

The system addressed many problems common to approximate optimization methods such as sensitivity to initializations and local minima. Although currently the system uses the gradient descent methods as the main optimization method, more sophisticated optimization systems can be plugged in for a more robust solution.

We validated the system by running experiments on real and synthetic images. The experiments showed us the robustness of our system against local minima and imposing constraints on regions where surface texture is not available. Overall, we are very encouraged with the results.

The current limitations of the system includes the inability to recover surfaces with very dense discontinuities. However, it is not very difficult to extend this system with a discontinuity preserving model. It should be noted that, even with a continuous smoothness energy, our system can perform as good as some of the systems in the literature that use discontinuity preserving models. Another problem with the current system is the handling of occlusions. Our system does not consider occlusions, which is not very realistic. The experimental results show that the occluded areas are handled by the smoothness terms, which produces acceptable results most of the time. However, in order to handle occlusions more effectively, we need to address this issue explicitly.

6 Acknowledgements

This work is supported by TUBITAK Career Project 105E097.

References

[1] Yusuf Sinan Akgul and Chandra Kambhamettu. Recovery and tracking of continuous 3d surfaces from stereo data using a deformable dual-mesh. In *International Conference on Computer Vision*, pages 765–772, 1999.

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222 – 1239, Nov 2001.

[4] T. Frohlinghaus and J. M. Buhmann. Regularizing phase-based stereo. In *International Conference on Pattern Recognition*, 1996.

[5] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(7):609–628, July 1990.

[6] Minglun Gong and Yee-Hong Yang. Fast unambiguous stereo matching using reliability-based dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(6), 2005.

[7] W. Hoff and N. Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136, February 1989.

[8] B.K.P. Horn. *Robot Vision*. The MIT Press, 1986.

[9] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(2):328–333, February 2006.

[10] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, July 2000.

[11] G. Li and S. W. Zucker. Stereo for slanted surfaces: First order disparities and normal consistency. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 617–632, 2005.

[12] Y. Ohta and T. Kanade. Stereo by two-level dynamic programming. In *International Joint Conference on Artificial Intelligence*, pages 1120–1126, 1985.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.

[14] J. Sun, Y. Li, S. B. Kang, and H. Y. Shum. Symmetric stereo matching for occlusion handling. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages II: 399–406, 2005.

[15] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *International Conference on Computer Vision*, pages 900–907, 2003.

[16] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.