Optimal Multi-Scale Matching

Michael S. Lew
Leiden Institute for Advanced Computer Science
Leiden University
2333 CA Leiden
Netherlands
mlew@cs.leidenuniv.nl

Thomas S. Huang
Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801
USA
huang@ifp.uiuc.edu

Abstract

The coarse-to-fine search strategy is extensively used in current reported research. However, it has the same problems as any hill climbing algorithm, most importantly, it often finds local instead of global minima. Drawing upon the artificial intelligence literature, we applied an optimal graph search, namely A, to the problem. Using real stereo and video test sets, we compared the A* method to both template and hill climbing. Our results show that A* has greater accuracy than the ubiquitous coarse-to-fine hill climbing pyramidal search algorithm in both stereo matching and motion tracking.*

1. Introduction

Stereo matching and motion tracking are two important areas in computer vision which involve finding correspondences in space (stereo) or time (motion/video). Stereo matching is one of the most flexible and general methods for recovering 3D structure, and is used in human-computer interaction and 3D modeling applications. Motion tracking is particularly useful for modeling elastic and rigid body motion. Analysis of facial gestures requires tracking points on the subject's face. Furthermore, it is also typically used in video analysis and compression. In video analysis, the camera motion can be extracted from the pixel motion within the camera shot. Also, in MPEG, inter-image compression is achieved by finding similar blocks between sequential images.

In order to accurately describe the stereo vision process, we define some relevant terms. L and R are two intensity images of overlapping content. Using a perspective projection camera model, a 3D point, (X,Y,Z) projects through the image planes of L and R at pixels (x_L,y_L) and (x_R,y_R), respectively.

A stereo matching algorithm typically involves the following steps:

(1) Find the correspondences between the stereo pair
(2) Refine each correspondence to subpixel accuracy
(3) Given the camera calibration, calculate the 3D position of each correspondence
(4) Reconstruct the surface given the 3D points

Correspondence based motion tracking is equivalent to stereo matching in the case where the 3D world is static, in which case the stereo baseline is the distance that the camera moved. In the stereo matching context, the most important constraint is that the correspondences must lie on the epipolar line. In the motion estimation problem, the epipolar constraint is not valid. A typical heuristic is that the motion is constrained to a local region around the pixel position.

This paper focuses on the problem of automatically finding correspondences using multi-scale or pyramidal matching. Significant previous work has been done in two view pyramidal matching. Beginning with Moravec[25] in 1977, image pyramids were used to obtain logarithmic computational efficiency in finding correspondences. In the 80s, the most relevant work was done by Hannah[5] whose algorithm also used multi-scale pyramids in conjunction with a hill climbing search algorithm. Dyer[2] gives an excellent overview of multi-scale methods as well as applications. The interested reader is referred to [1,4,5,6,7,9,17,18,20,22,23,24]. Recently, there have been methods which examined the noise distribution[12] and texture correlation[13]. Both of these methods also used coarse to fine hill climbing search algorithms.

1063-6919/99 $10.00 (c) 1999 IEEE

This paper shows that matching image pyramids is equivalent to solving the minimum cost path problem for a tree. In Section 2, we briefly introduce and discuss multi-scale matching followed by the relevance of A*[8,11], which finds the optimal path unlike the hill climbing method which can stop at local minima. Section 3 describes our experiments and we give conclusions in Section 4.

2. Multi-Scale Matching

What is an image pyramid? An image pyramid is a collection of copies of the original image at different sizes. These sizes represent varying levels of scale. For the purposes of this paper, we assume that the sizes of the image copies are at powers of 2 (i.e. 1x1, 2x2, 4x4, 8x8, etc.), which has also been called a 1:2 pyramid. The typical method is to blur the original image with a Gaussian filter[15] and then subsample to create the lower resolution level. One of the important advantages of the Gaussian method is that copies can be synthesized at any scale factor due to the sigma parameter in the Gaussian blurring.

2.1 Coarse-to-Fine Matching: An Example

The purpose of this example is to illustrate the process of finding a pixel correspondence between two images called L and R. Also, let us assume that the coordinates are described in (row, column, level) format for L and R. In the example shown in Figure 1, the selected pixel in the left image is on level 2 which is denoted as L(2,0,2). By definition, we also know that the selected pixel also maps to left image level 1 at L(1,0,1). The coarse-to-fine search process begins with an initial match state at the coarsest level available, which in our example is level 1. This initial correspondence can be found by either starting at the 1x1 level or using a different search process to find the initial match state. In Figure 1, assume that the initial match state in level 1 from the left to right image is [L(1,0,1) -> R(1,0,1)]. From R(1,0,1), there are 4 possible candidates in the right image on level 2, namely, R(2,0,2), R(2,1,2), R(3,0,2), and R(3,1,2).

In the case of stereo matching where images have been rectified toward the epipolar constraint, then we can eliminate R(3,0,2) and R(3,1,2), which is why they are shown with dotted lines. The decision between R(2,0,2) and R(2,1,2) is made by comparing the local error or cost, C(L(2,0,2), R(2,0,2)); and C(L(2,0,2), R(2,1,2)). For simplicity sake, we can write the expression without L and say we take the lesser of C(R(2,0,2)) and C(R(2,1,2)) as the correspondence.

2.2 Pyramids and Trees

The pyramidal search problem is precisely a path minimization problem in graph theory. Each image pyramid is a tree in which each pixel at level n has four children at level n+1. With the epipolar assumption, the number of children is reduced to 2 at level n+1. Let us define the cost of traversing path R(a,b,n-1) to R(r,c,n) relative to L(x,y,n) as C(R(r,c,n)) as shown in Figure 2. Note that it is not necessary to specify R(a,b,n-1) since R is a tree, which implies that R(r,c,n) can only have one parent. Furthermore, there is an implicit assumption that C is comparing L(x,y,n) with R(r,c,n). However, it is redundant to write L(x,y,n) because it is fixed relative to the selected pixel in L. In this framework, C is a binary tree of depth N-1, where N is the number of levels in R.

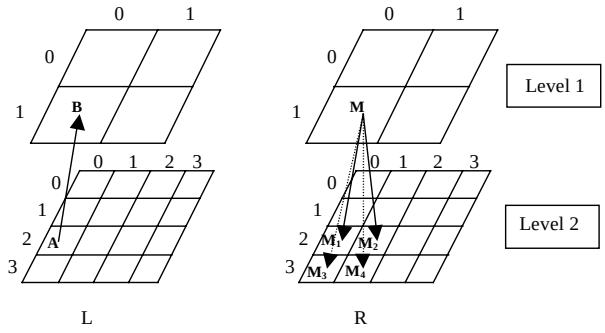

Figure 1. The pyramidal matching process between stereo images. **A** is user selected. **B** is the projection of **A** in the left pyramid. **M** is the pixel which has the least feature error to **B**. **M** has four children $\mathbf{M_1}$, $\mathbf{M_2}$, $\mathbf{M_3}$, and $\mathbf{M_4}$, which are candidates to match to **A.** The child which has the least error would be the correspondence of pixel **A**.

In the simplest coarse-to-fine matching algorithm, we being at the 2x2 resolution, select the match with minimum error, and propagate to 4x4, 8x8, etc. This algorithm is both intuitive and elegant and leads to O(logN) time, where N is the resolution of the original image. Using such a method led to an accuracy of 0.51 on the Stuttgart image[3,5,14] database.

Why was the coarse-to-fine accuracy so low? The problem was that the matching algorithm can make errors in propagating the coarse scale match to a finer scale match. One solution is to begin the matching algorithm at a finer scale level. In the literature [5,9], 16x16 and 32x32 are often used as the initial scale level.

2.3 The A* Graph Search Algorithm

How can we improve the accuracy? To improve the accuracy, we need to examine why the coarse-to-fine

matching algorithm is failing. On consideration of the algorithm, we see that it is intrinsically a hill-climbing algorithm. It selects the minimum error path locally at each node and does not keep track of other paths which might lead to globally optimal solutions. This means that it is also prone to all of the faults of any hill climbing algorithm, most importantly, it can stop at local minima. How can we improve the matching algorithm? By applying a more sophisticated matching algorithm such as A*[8,10,11], we can prove that we have found the global minimum instead of just a local minimum with regard to the total path traversal cost as shown in Figure 3.

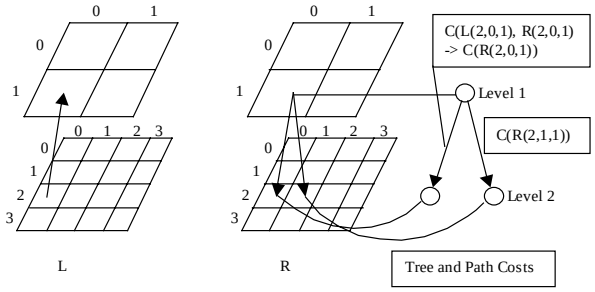

Figure 2. Pyramidal search relating to graph search in trees

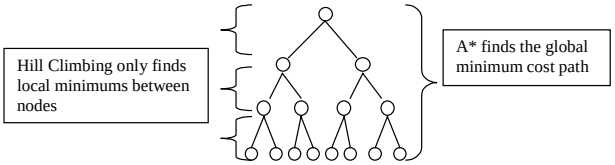

Figure 3. Comparing Hill Climbing to A*

The A*[8,11] algorithm can be described as follows:

Step (1) Initialize Q to a zero-length path that only contains the root.
Step (2) P = first path of Q
While (P is not a leaf)
begin
  Step (3) S = list of new paths created by children or successors of P
  Step (3.1) Delete all new paths in S with loops
  Step (4) Calculate costs of new paths in S and append to Q.
  Step (4.1) If two or more paths in Q reach a common node, remove all paths except the one which reaches the common node with the minimum cost.
  Step (5) Sort Q by the sum of the path cost and the estimated cost to reach a leaf (heuristic)
  Step (6) P = first path of Q
end

These are some observations regarding the use of A* on searching image pyramids. Since the graph formed from the image pyramids is a tree, it is not necessary to perform Step (3.1) because trees are acyclic. Step (4.1) is not necessary because each child node only has only one parent in a tree, which means that it is not possible for two paths to reach a common child node.

It has been previously proven that if the heuristic is a lower bound estimate on the actual distance then A* produces globally optimal paths[8]. Therefore in step (5), the estimated cost to reach a leaf from the current path is set to zero in order to guarantee optimality.

Initialization of Q is to the root of the pyramid, which is the coarsest scale level. P is regarded as the current best candidate. If P is at the finest level, we have found the minimum cost match and we STOP. Otherwise, we generate the children of P and the costs to reach the children. The children are the pixels at the next finer image scale level, which contributed to the genesis of P in creating the image pyramid.

3. Experiments

For the stereo matching tests, we used 4 stereo image pairs: Poster, Rock Wall, Street, and Robot. In each case the query pixels in the left image were found automatically by selecting only those pixels which had a gradient magnitude of intensity greater than a threshold. The ground truth correspondences in the right image were found manually. The poster image dataset of 8,216 query pixels represents matching on a plane in 3D, with only the perspective and lens distortion. The Rock Wall stereo dataset of 2,281 query pixels was from the difficult category of the Stuttgart[3,5,14] standardized stereo image set. The Stuttgart standardized set had three levels of matching difficulty: simple, moderate, and difficult, and was the basis of a large comparative survey of stereo matching methods[14] conducted by ISPRS Working Group III/4. The Street stereo database of 439 query pixels represents a street with pavement, trees, and grass. This would be indicative of performance regarding automatic highway/city mapping and computer based automobile driving. The Robot stereo database of 1,275 query pixels contains two industrial robot arms which would be representative of construction or industrial settings.

The goal of the motion tracking experiments was to find the 2D pixel correspondences over a video sequence. For the video motion tracking experiments we used four video clips from two sources, namely, the movie, ”Four Weddings and a Funeral” and the JonFace sequence from University of Illinois[26]. The ground truth for “Four Weddings and a Funeral” was obtained from Philips Research, Eindhoven. We used 3 clips (120 frames each,

each frame with 256 manually marked pixels), which involved (1) camera movement only (the initial shot of the friends sleeping) (2) a moving vehicle (the friends driving off to the mansion); and (3) multiple people moving (the first wedding reception).

In the JonFace testset, the ground truth for the frame-to-frame point matching was determined by placing colored markers about 3 pixels in diameter on the subject's face, recording an image sequence, and using correlation to find the positions of the marks. For the experiments, a method using thresholding and bilinear interpolation was devised to remove the markers[26]. The resulting sequence is composed of 20 frames. Example images in the Jonface sequence are shown in Figure 4.

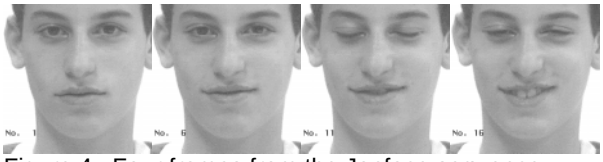

Figure 4. Four frames from the Jonface sequence.

3.1 Implementation

Implementation required choices of a feature class, a feature metric, and a multiscale representation for use in the hill climbing and A* algorithms. Normalized intensity was chosen as the feature class due to good results by several researchers[19,20,21]. Recent work[16] has indicated that from a maximum likelihood perspective, the noise is closer to exponential than Gaussian in stereo matching, which in turn suggests that the sum of the absolute difference would be a better feature metric.

The multiscale representation was an image pyramid generated by blurring the originals with a Gaussian and then subsampling by a factor of 2, repeatedly until the image copy was of size 2x2. For both the A* and the hill climbing search algorithms, we used 3x3 windows for the comparisons. The single scale template matcher used a fixed window of size 9x9 on the finest scale level. On a Pentium 120 CPU computer, the time required to match each pixel was between 0.001 and 0.014 seconds for the hill climbing algorithm and less than 0.002 seconds for the A* algorithm.

3.2 A Visual Example of Hill Climbing Versus A*

In this section we give a visual example of the matching results between the hill climbing algorithm versus A* for a standard stereo pair. The ground truth correspondences for the Rock Wall stereo pair are shown in Figure 5 and the accuracy (percentage of matches within 1 pixel of the ground truth) results for the hill climbing and A* algorithms are shown in Table 1. Notice that the hill climbing method peaks with an accuracy of 76 and that A* achieves 88 percent. Figure 6 gives a visual impression of the number and positions of the incorrect matches from the hill climbing and A* algorithms. The ideal result would be a white image.

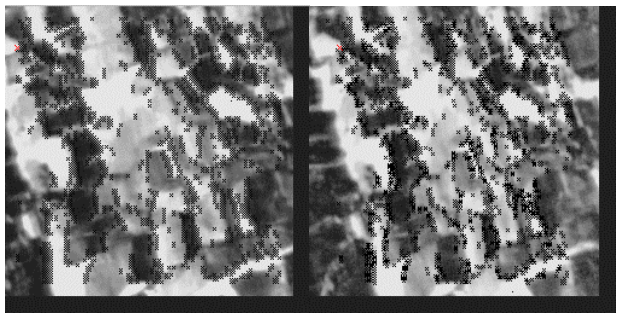

Figure 5. Spatial positions of 2,281 ground truth correspondences on the Rock Wall stereo pair shown as crosses (we used crosses because pixels are sometimes difficult to differentiate from the image itself on B/W media).

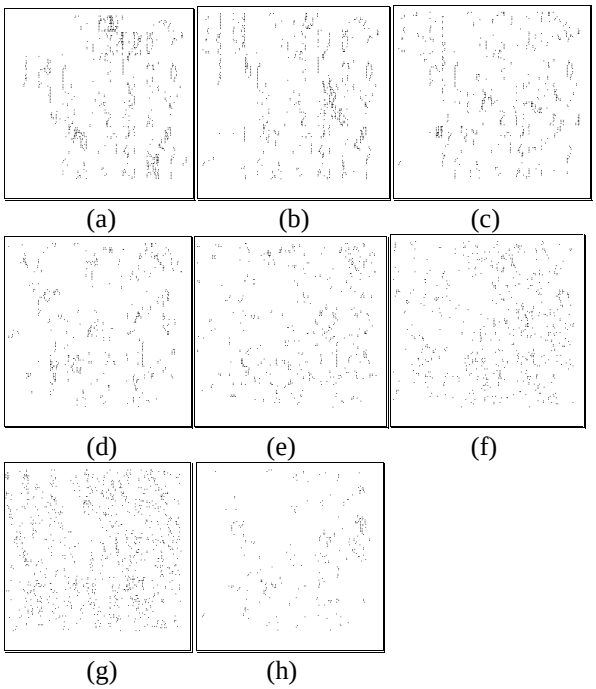

Figure 6. Positions of incorrect matches in the right image of the Rock Wall stereo pair using algorithms hill climbing (HC) and A*. HC with start level 4x4(a); 8x8(b); 16x16 (c); 32x32(d); 64x64(e); 128x128(f); 256x256(g); and A*(h).

3.3 Results from Stereo Matching Test Sets

The comparative results between the hill climbing, A* algorithm and the template matcher algorithms are

shown in Table 2. For these experiments, the template matcher had a matching accuracy of 71.2%. The hill climbing method reached a maximum accuracy of 80.6% at starting level 32x32 while A* had fewest mismatches with an accuracy of 94.2% as shown in Table 2.

Table 1. Accuracy For Hill Climbing(HC) And A* On The Rock Wall Stereo Pair

Method	Accuracy %
HC, Starting Scale 4x4	47
HC, Starting Scale 8x8	54
HC, Starting Scale 16x16	65
HC, Starting Scale 32x32	73
HC, Starting Scale 64x64	*76*
HC, Starting Scale 128x128	70
HC, Starting Scale 256x256	53
*A**	*88*

Table 2. Average Stereo Matching Accuracy For Hill Climbing(HC) And A* On A Gaussian Pyramid

Method	Accuracy % (average)
HC, Starting Scale 4x4	29.3
HC, Starting Scale 8x8	48.6
HC, Starting Scale 16x16	62.0
HC, Starting Scale 32x32	*80.6*
HC, Starting Scale 64x64	78.2
HC, Starting Scale 128x128	76.3
HC, Starting Scale 256x256	59.6
Template: Template Size 9x9	*71.2*
*A**	*94.2*

3.3 Results from the Motion Tracking Test Sets

Table 3 displays the results for the hill climbing algorithm, template, and the A* algorithm in the area of motion tracking. Note that the hill climbing method peaks at starting scale 64x64 as opposed to 32x32 in the stereo matching test sets. Furthermore, the template matching method performed comparably with the hill climbing method at the optimal starting scale. The A* algorithm had a matching accuracy of 92.7 as compared to the peak accuracy of 84.2 for the hill climbing algorithm

3.4 Discussion

In this paper, we addressed the problem of finding correspondences between stereo pairs. We examined the problem of matching image pyramids and showed that it was equivalent to a minimum cost path problem from graph theory. Within this graph theory context, it was observed that the traditional method of solving the minimum cost path problem was equivalent to a hill climbing search algorithm. Hill climbing algorithms have the important deficiency that they can stop at local minima. Drawing upon the artificial intelligence literature, we applied an optimal graph search, namely A*, to the problem.

Table 3. Average Motion Tracking Accuracy For Hill Climbing(HC) And A* On A Gaussian Pyramid

Method	Accuracy % (average)
HC, Starting Scale 4x4	31.7
HC, Starting Scale 8x8	46.4
HC, Starting Scale 16x16	59.1
HC, Starting Scale 32x32	*76.5*
HC, Starting Scale 64x64	84.2
HC, Starting Scale 128x128	70.6
HC, Starting Scale 256x256	51.3
Template: Template Size 9x9	*85.1*
*A**	*92.7*

From the experimental results, A* clearly beats hill climbing in accuracy and is comparable in computational efficiency depending on the starting level for the hill climbing method.

The first topic which we address is why A* had consistently greater accuracy than hill climbing. When hill climbing makes a bad choice of directions in the coarse to fine search, it can never backtrack no matter how large the error becomes at the fine scale levels. A* is providing an automatic backtracking mechanism for choosing a new direction at a coarser scale level if the accumulated error is greater than another path.

Second, we discuss generality of the A* approach. Specifically, is A* limited to the feature class, metric, or pyramid representation used in these experiments? In our setup, the feature class and feature metric define the path cost between levels in the pyramid. This implies that any feature class and metric could be utilized within the A* algorithm. Furthermore, A* can be used with other pyramid representations such as Laplacian or averaging pyramids.

Third, A* provides a general framework for the inclusion of heuristics into the cost function. Heuristics allow the stereo matcher to be optimized for particular applications and integrated with other kinds of knowledge. For example, if we have performed apriori segmentation on the images into constituent regions and know for each region, the size, shape, color, and texture, then we can bias the stereo matcher to favor regions having similar size and shape, while penalizing regions which have differing colors or textures.

4. Conclusions

In this paper, we examined the usage of a globally optimal search algorithm, A*, versus the traditional hill climbing algorithm. The A* algorithm had significantly greater accuracy than the hill climbing algorithm in both stereo matching and motion tracking.

The A* refinement to the image pyramid matching algorithm can be extended to any feature class, feature metric or image pyramid representation. In particular, we think that most hierarchical computer vision algorithms or significant parts therein could be reformulated as graphs within a minimum cost path traversal context, and therefore be solved using A* instead of hill climbing.

Future work will be focussed on integrating heuristics from higher level primitives into the matching algorithms for both stereo and motion. The inclusion of these primitives is already part of the framework of the A* algorithm.

Acknowledgements

This research was funded by the Dutch National Science Foundation. We would like to thank Philips Research, Eindhoven for lending us their dataset for "Four Weddings and a Funeral."

References

[1] Cohen, L., L. Vinet, and P. Sander, "Hierarchical Region Based Stereo Matching," *Proc. of Computer Vision and Pattern Recognition*, pp. 416-421, 1989.

[2] Dyer, C. R., "Multiscale Image Understanding," **Parallel Computer Vision**, Academic Press, Inc., Orlando, Florida, 1987.

[3] Forstner, W., Institute for Photogrammetry, University of Stuttgart, West Germany, 1986.

[4] Grimson, W. E. L., *From Images to Surfaces*, M.I.T. Press, Cambridge, MA, 1981.

[5] Hannah, M. J., "Digital Stereo Image Matching Techniques," *ISPRS*, Kyoto, vol. 27, 1988.

[6] Hoff, W. A. and N. Ahuja, "Depth From Stereo," *Proc. of Fourth Scandinavian Conference on Image Analysis*, June 18-20, Trondheim, Norway, pp. 761-768, 1985.

[7] Lim, H. S. and T. O. Binford, "Stereo Correspondence: A Hierarchical Approach," *Proc. of Image Understanding Workshop*, February, 1987.

[8] Nilsson, N. J., *Principles of Artificial Intelligence*, Tioga Publishing Company, 1980.

[9] Weng, J., N. Ahuja and T. S. Huang, "Two-View Matching," *Proc. of Second International Conference on Computer Vision*, pp. 64-73, 1988.

[10] Hart, P.E, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Path," *IEEE Trans. on SCC*, 4:100-107, 1968.

[11] Rich, E., and K. Knight, **Artificial Intelligence**, McGraw-Hill, pg. 75-76, 1991.

[12] Menard, C. and A Leonardis, "Robust Stereo on Multiple Resolutions," *International Conference on Pattern Recognition*, August 25-29, pg. 910-914, 1996.

[13] O'Neill, M. and M. Denos, "Automated System for Coarse-to-Fine Pyramidal Area Correlation Stereo Matching," *Image and Vision Computing*, vol. 14 (3), pp. 225-236, 1996.

[14] Gulch, E., "Results of Test on Image Matching of ISPRS WG III/4," *International Archives of Photogrammetry and Remote Sensing*, vol. 27-III, pp. 254-271, 1988.

[15] Haralick, R. M., and L. G. Shapiro, **Computer and Robot Vision**, Addison-Wesley, Reading, Massachusetts, 1993.

[16] Sebe, N, M. Lew and N. Huijsmans, "Which Ranking Metric is Optimal? With Applications in Image Retrieval and Stereo Matching," *Proc. of International Conference On Pattern Recognition*, Brisbane, Austrailia, August, 1998.

[17] Aschwanden, P. and W. Guggenbuhl, "Experimental Results from a Comparative Study on Correlation-type Registration Algorithms," **Robust Computer Vision** (edited by W. Forstner and S. Ruwiedel), pp. 268-289, 1992.

[18] Dhond, U. R., and J. K. Aggarwal, "Structure from Stereo- A Review," *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1489-1510, Nov-Dec., 1989.

[19] Kanade, T., "Development of a Video-Rate Stereo Machine," *Image Understanding Workshop*, Morgan Kaufman Publishers, Monterey, CA, November, pp. 549-557, 1994.

[20] Matthies, L. H., R. Szeliski, and T. Kanade, "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences," *Int. Journal of Computer Vision*, 3:209-236, 1989.

[21] Scharstein, D., and R. Szeliski, "Stereo Matching with Non-Linear Diffusion," *International Journal of Computer Vision*, 28(2):155-174, 1998.

[22] Kanade, T., and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9):920-932, 1994.

[23] Grimson, W.E.L., **From Images to Surfaces: A Computational Study of the Human Early Visual System**, MIT Press. Cambridge, MA, 1981.

[24] Terzopoulos, D., "The Computation of Visible-Surface Representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4), pp. 417-437, 1988.

[25] Moravec, H.P., "Towards Automatic Visual Obstacle Avoidance," *Proc. 5th IJCAI*, August 1977.

[26] Tang, L., Y. Kong, L. S. Chen, C. R. Lansing, and T. S. Huang, "Performance Evaluation of a Facial Feature Tracking Algorithm," *Proceedings of the NSF/ARPA Workshop: Performance vs. Methodology in Computer Vision*, pp. 218-229, 1994.