A Comparison of Iterative Bayesian Methods for Solving the Sparse Point Correspondence Problem.

Markus Louw *Fred Nicolls* *Dee Bradshaw*

Department of Electrical Engineering
University of Cape Town, South Africa
markus.louw@gmail.com

Abstract

This paper describes two iterative Bayesian methods for solving the sparse point correspondence resolution problem. This problem arises whenever there are conflicting point labellings resulting from the use of a simple region based match measure between points in different images. The first is a Loopy Belief Propagation method, the second an iterative method first applied to the dense stereo problem. Both algorithms are simple in the sense that we have not used a criterion which includes a priori shape information or relative orientation information, only that the distance between points and neighbours of points should be preserved. The correspondence resolutions generated by these algorithms are of a similar quality. Ground truth point sets generated by sampling from the Middlebury stereo image sets [3] were used to compare the algorithms.

1. Introduction and Literature Review

The problem of sparse stereo correspondence is to match each feature point in a source image to its corresponding feature point in a second image. We encounter this problem in a wide range of computer vision applications, e.g. scene object recognition [1], target tracking (where points on the target are tracked) and sparse 3D reconstruction [11]. Various assumptions can be made, just as within the dense stereo matching paradigm, about the spatial relationships between points in each set, and how these should affect the matching process. The simple initial step of applying a correlation match measure to candidate matches and assigning normalized probabilities for each point to its candidate matches usually results in matching incompatibilities, and ignores useful information about the relative orientations and neighbourhood structures of the points.

Perhaps the first description of the problem was given by Ullman [14], in which three principles for matching are given as the *principle of similarity*, the *principle of proximity*, and the *principle of exclusion* (final correspondences between source and target points must be one to one). Some previous attempts at resolving these incompatibilities include the ”winner take all strategy” of [8], the ”some winners take all” of [15], an SVD proximity matrix approximation [10], and resolution by estimation of approximate affine transformations between matches in [4]. In [12], maximal cliques in a relational subgraph are established, and in [5] a concave programming approach was used.

In [6], the problem is cast into a Maximum Weight Maximum Cardinality graph matching problem and is solved using a push-relabel algorithm. In [13], a Bayesian model selection paradigm is used to simultaneously find point correspondences and a model which describes the transformation between the points in the stereo pair. More relevant to our experiments is [9] (or [2] for a more detailed description of the algorithm), in which Perwass et al suggested an iterative Bayesian technique for solving the dense correspondence problem, which we here adapt to solve the problem of sparse point correspondence resolution. We apply and test this adapted algorithm, and also test a Loopy Belief Propagation (LBP) approach, thus allowing us to compare the two.

2. Neighbours and candidates

Suppose we are given two point sets X_1 and X_2, in the first and second images respectively. For each i_{th} point $X_1(i)$ in X_1, we search through X_1 to find neighbouring points (within a certain radius), and form a vector of neighbour indices $N_1(i)$ for this point. So if two points with indices j and k are within a certain radius of a point with index i in X_1, we would have $N_1(i) = [\ j\ k\ ]$. We form N_2 in the same way, using points in the second image.

Similarly, for each i_{th} point $X_1(i)$ in X_1 we make a list $C(i)$ of candidate points in X_2 for that point. We overload the notation in such a way that if $C(i)$ refers to candidate matches in X_2 for point $X_1(i)$, and if there are $n(C(i))$ number of candidate matches we may refer to the j_{th} such point as $C(i, j)$. Similarly we want to refer to the j_{th} neighbour of $X_1(i)$ as $N(i, j)$. This scheme is depicted in Fig. 1.

To see a real example of feature points and their candidate points see Fig. 2, with the lines shown between each source point and its candidate target points. Similarly, neighbours are connected by lines and shown in Fig. 3 and Fig. 4. In Fig. 4 there are more points and consequently more neighbours and more candidate matches.

3. Resolving sparse correspondences using Loopy Belief Propagation

We form a loopy Bayesian network, where the variable nodes represent points in X_1, and each point $X_1(i)$ is connected to each of its neighbours in $N(i)$. Each variable node $X_1(i)$ takes on a state vector of probabilities with the k_{th} element in this state vector representing the probability of correct assignment/matching of point $X_1(i)$ to point $C(i, k)$ in X_2. The state vector for each point node is initialized with the normalized outputs a simple region matcher (such as 2D window correlation).

The following equations express the Loopy Belief propagation algorithm using factor nodes (following loosely the notation of [7]):

$$\mu_{x \to f}(x) = o(x) \prod_{g \neq f} \mu_{g \to x}(x) \qquad (1)$$

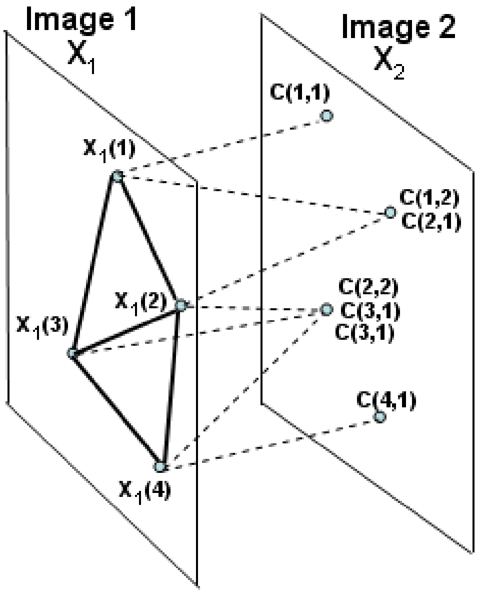

Figure 1: This figure shows the neighbourhood and candidate schemes for X_1 in the first image. The quadrilateral on the left is the first image, containing four points, each labelled as $X_1(i)$ with $i = 1..4$. The points within a radius of eachother are neighbours (joined by thick lines), and each point's candidates in the second image (the quadrilateral on the right) shown by connectivity with a dotted line. To avoid cluttering, the neighbourhood system of points in the second image has not been delineated.

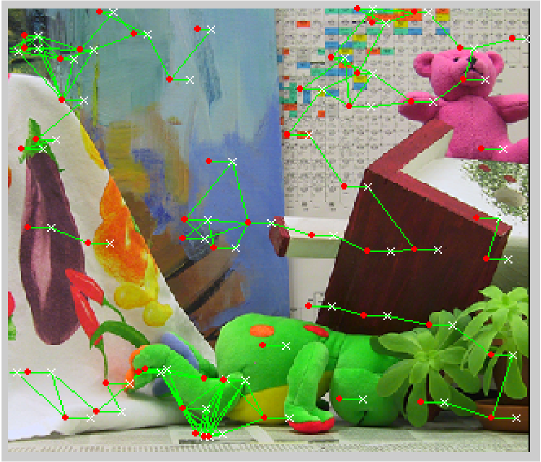

Figure 2: Example of feature points (crosses) and their candidates (dots) superimposed on the first image of the image pair.

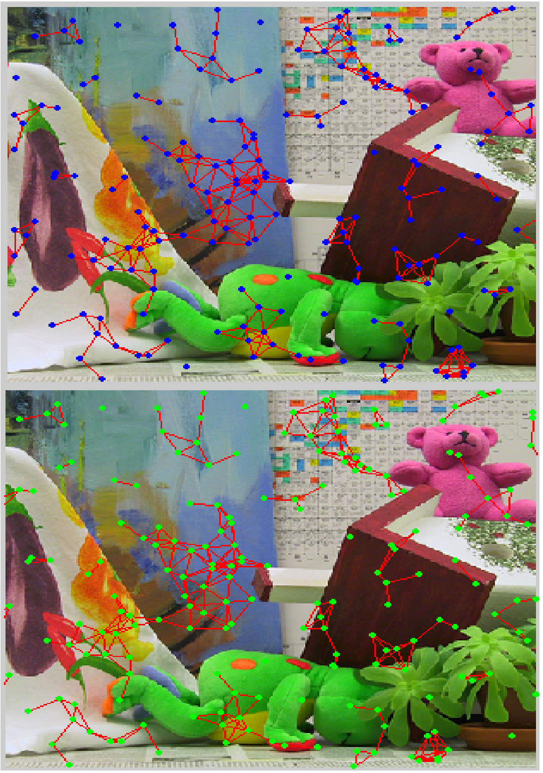

Figure 3: The top frame and bottom frames are the first and second images respectively. Feature points and neighbourhood systems are shown for each point set in each image. There are 200 feature points in each image.

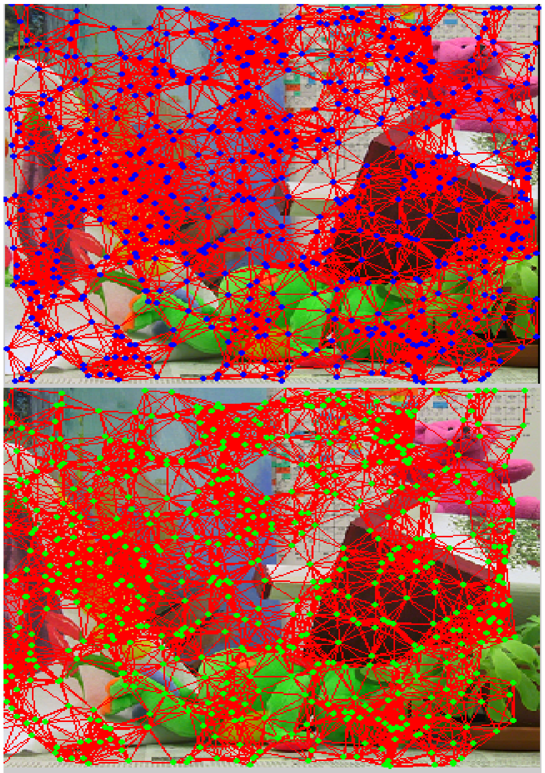

Figure 4: The top frame and bottom frames are the first and second images respectively, with neighbourhood systems for each frame shown. In this example there are 600 points, so the average number of neighbours per point (ANN) increases. The average number of candidates per point (ANC) would also increase, but this is not shown.

$$\mu_{f\to x}(x) = \sum_{u\setminus x} f(u) \prod_{y\neq x} \mu_{y\to f}(y) \tag{2}$$

Where x, y are variable nodes, g, f are factor nodes, $\mu(.)$ is a message from a variable to factor node or vice versa, and $o(x)$ is evidence on the variable node x. In a Loopy Belief Propagation scheme the above two equations are iterated usually until convergence. In our experiment, we used the max-product update algorithm.

This framework can be used to resolve correspondences between sparse points on image pairs. Seen in terms of a Markov Random Field (MRF) energy function on the target index assignments for points in X_1, we can describe the joint probability distribution over points and pairwise neighbours as,

$$p(C|Y) \propto \prod_k \psi_k(c_k, y_k) \prod_{k,l:k<l,l\in N_1(k)} \psi_{kl}(c_k, c_l) \tag{3}$$

where C refers to the correspondence labelling of each point in X_1, Y is all the observation information on the candidate match strengths for each point in X_1, y_k is the candidate location and image information, $N_1(k)$ are the neighbours of node k, $\psi_k(c_k, y_k)$ is the energy on a particular set of candidate matches for a point, also known as the local evidence (in our case found by a modified window correlation matcher), and $\psi_{kl}(c_k, c_l)$ is the energy on a pair of neighbouring points with particular labellings. The energy for a point match is given by

$$\psi_k(c_k, y_k) \propto \exp\big(-s\big(X_1(k), C(X_1(k), c_k)\big)\big) \tag{4}$$

where $s(\cdot,\cdot)$ is a matching function between points across images. The potential function involving point pairs is described as

$$\psi_{kl}(c_k = i, c_l = j) \propto$$
$$\begin{cases} \exp\big(-|\mathbf{dist}\big(X_2(C(k,i), X_2(C(l,j))\big)\big) - \cdots \\ \qquad\qquad \mathbf{dist}\big(X_1(k), X_1(l)\big)|\big) \\ \qquad\qquad \text{if } X_2\big(C(k,i)\big) \neq X_2\big(C(l,j)\big) \\ z \qquad\qquad \text{if } X_2\big(C(k,i)\big) = X_2\big(C(l,j)\big) \end{cases}$$

In the above equation, the arguments in $\psi_{kl}(c_k = i, c_l = j)$ indicate that point $X_1(k)$ is matched to $C(k,i)$ and $X_1(l)$ is matched to $C(l,j)$, (and this is the energy function for this joint hypothesis); **dist(·,·)** is a measure of the Euclidean distance between the two points indexed by its arguments. The z energy term is some small number greater than zero. (We used $z = 0.01$. If this value is forced to zero, match possibilities are immediately excluded by the LBP algorithm, and it does not converge correctly). After convergence of LBP to minimize the MRF energy, we have an a posteriori estimate for $p(C|Y)$, from which we can take the MAP label for each point as its correct candidate.

4. Iterative Beyesian method applied to sparse stereo correspondence

Given an assumed pixel match $(X_1(A), X_2(C(A,i))$ and a particular neighbour $X_1(B)$ in the set $N(X_1(A))$ (which is the set of neighbours of $X_1(A)$), we define a compatibility function $h(X_1(A), X_2(C(A,i), X_1(B), X_2(C(B,j)))$ which gives the a priori probability for $X_2(C(B,j))$ being a correct match for $X_1(B)$. This matching function $h(\cdot,\cdot,\cdot,\cdot)$ has the same role as the compatibility energy function (used to compute the outgoing factor node messages) in the previous section. To save space (and incidentally to keep it more similar to the appearance of equations in [9]), we identify $x_A = X_1(A)$, $y_A = X_1(B)$, $x_B = X_2(C(A,i))$ and $y_B = X_2(C(B,j))$. The probability of (x_A, x_B) and (y_A, y_B) being two neighbouring point matches is

$$p\big(X_B = x_B, Y_B = y_B | A, B, X_A = x_A, Y_A = y_A\big) = s(x_A, x_B)s(y_A, y_B)h(x_A, x_B, y_A, y_B) \quad (5)$$

with

$$\hat{y}_B = \text{argmax}_{y_B}\left(\frac{P(X_B, Y_B = y_B | A, B, X_A, Y_A)}{\max_y P(X_B, Y_B = y | X_A, Y_A)}\right).$$

If, for a particular set (x_A, x_B, y_A), the corresponding prior is maximized by $\hat{y}_B$, we have

$$P\big(X_B = x_B, Y_B = \hat{y}_B | A, B, X_A = x_A, Y_A = y_A\big) = s(x_A, x_B)s(y_A, y_B). \quad (6)$$

We would like to incorporate neighbourhood match likelihoods in the calculation of the candidate match probabilities for any particular point, i.e. any point's candidate match probabilities should depend on its neighbours matching probabilities. Thus we take our final match probability estimate for a point pair (x_A, x_B) to be

$$P\big(X_B = x_B | A, B, X_A = x_A\big) = \rho \cdot s(x_A, x_B) \sum_{k=1}^{n(x_A)} \max_i s(N(x_A, k), N(x_B, i)) \cdots \frac{h(x_A, x_B, N(x_A, k), N(x_B, i))}{\max_j h(x_A, x_B, N(x_A, k), N(x_B, j))'} \quad (7)$$

where ρ is a normalizing factor, $n(x_A)$ is the number of neighbours of the point x_A. This function is clearly iterative, as we may calculate a new set of probabilities every time a neighbour's match probabilities are updated. Through the function $h(x_A, x_B, y_A, y_B)$ we may assert a similar compatibility rule as was done in the previous section, by specifying:

$$h(X_1(A), X_2(C(A,i), X_1(B), X_1(C(B,j))))) = \begin{cases} \exp\big(-|\textbf{dist}\big(X_2(C(A,i), X_2(C(B,j))))\big) - \cdots \\ \qquad \textbf{dist}\big(X_1(A), X_1(B)\big)|\big) \\ \qquad \text{if } X_2(C(A,i)) \neq X_2(C(B,j)) \\ 0 \qquad \text{if } X_2(C(A,i)) = X_2(C(B,j)) \end{cases}$$

5. Match measure

When calculating the match measure for the term $s(x_A, x_B)$, there is a range of different possibilities, e.g. 2D-correlation, Kullbeck-Leibler divergence, Mutual Information, Earth Mover's Distance, etc. Each of these would operate better under different conditions. In this experiment, we use a modified 2D correlation function which returns values in the range [0,1], where a high value indicates a good match. Since we are using colour images, the correlations per colour channel are averaged to return the match measure $s(x_A, x_B)$ for a point pair.

N	Max corr	LBP	Perwass	ANN	ANC
50	87.6	87.3	87.1	0.5	0.5
80	81.7	85.5	85.1	0.85	0.86
100	78.5	86.1	85.6	1.1	1.1
120	74.8	86.8	85.8	1.3	1.3
150	71.1	87.6	86.2	1.6	1.6
200	66.0	89.3	87.3	2.2	2.2
300	56.9	91.5	86.9	3.3	3.3
400	50.5	88.8	67.7	4.3	4.3

Table 1: Table of match results for small candidate and neighbour search windows (25 pixels).

N	Max corr	LBP	Perwass	ANC	ANN
50	71.5	88.6	87.5	2.0	2.1
80	62.2	91.3	88.9	3.2	3.3
100	58.2	96.1	91.1	4.1	4.2
120	53.8	97.7	93.0	4.9	4.9
150	48.9	98.0	92.5	6.2	6.2
200	44.1	98.0	92.0	8.2	8.3
300	29.1	99.5	92.6	12.4	12.4
400	31.5	99.2	92.8	16.6	16.7

Table 2: Table of match results for larger candidate and neighbour search windows (50 pixels).

6. Results

The method we used to test this correspondence resolving algorithm was to use the ground truth dense stereo pairs from the Middlebury data set, which may be freely downloaded and is documented in [3]. One of the image sequences was chosen from this set, and random points were chosen from the first image. The ground truth disparity map was then used to find the correct corresponding points in the second image. The Middlebury [3] ground truth pairs, which were calculated using a structured lighting approach, provides an acceptable method to test sparse correspondence resolution algorithms, since by sampling randomly from points in the first image and using the ground truth disparity map to derive the corresponding second point set in the second image, we have a correct labelling for every random point set thus generated. The correspondences calculated by each of the algorithms is compared to this set of correct labellings, and a "percentage correct matches" statistic for each method may be derived. This is seen in the columns for the "Max corr", "LBP" and "Perwass" methods in Tables 1, 2 and 3.

The experimental results obtained from applying these al-

N	Max corr	LBP	Perwass	ANC	ANN
50	62.6	95.3	92.1	3.9	4.0
80	53.9	96.8	92.0	6.1	6.2
100	50.0	98.0	91.9	7.7	7.8
120	45.7	99.0	91.0	9.3	9.4
150	41.5	99.2	89.8	11.7	11.8
200	35.1	97.3	88.1	15.2	15.2
300	28.2	96.7	88.0	23.1	23.2
400	25.3	97.4	88.1	30.6	31.2

Table 3: Table of match results for largest candidate and neighbour search windows (70 pixels).

gorithms to point sets of varying sizes are shown in Tables 1, 2 and 3. In Table 1, the search window for neighbours and candidates was small (25 pixels), in Table 2 the search window was larger (50 pixels), and in Table 3 it was the largest (70 pixels), and the statistics vary accordingly. In these tables, N is the number of points, ”Max corr” is the unresolved correspondence set obtained by taking the maximum a posteriori (MAP) correlation estimate for each point’s candidates, ”ANN” is the average number of neighbours per point, and ”ANC” is the average number of candidates per point, over 400 runs. To see an example of the point correspondences generated by each of these methods, see Fig. 5.

6.1. Discussion

The statistics in the top half of Table 1, where a candidate and neighbour search window size of 25 pixels was used, show us that when the average number of candidates per point exceeds one, both iterative Bayesian methods improve the percentage of correct matches significantly (when ANC is below 1 there are obviously not many match conflicts to resolve). As the number of points in the scene increases, the correspondence set taken by choosing the MAP candidate for each point in X_1 (listed as the ”Max corr” statistic) worsens to only 50% correct matches, but the iterative methods improve this to about 88%. In Table 2 where the ANC and ANN averages increase due to the use of a larger window size (50 pixels), the ”Max corr” statistic worsens further. The iterative methods however actually improve, despite the increased averaged number of candidates per point (ANC). One may have expected the iterative methods to perform worse, since with more candidates there must surely be greater chance of mislabelling the points. In fact, when there are more neighbours as well as candidates (ANN and ANC are both greater when more points are involved), the iterative methods exploit the increase in neighbourhood and distance information, and resolve the points more successfully. This trend is extended in Table 3 where the ANN and ANC are higher still. The ”Max corr” measure worsens, but the iterative Bayesian methods achieve higher correct match statistics.

7. Conclusion

Both the above methods give a substantial improvement in the number of correct matches, particularly when there are higher numbers of candidate matches and point neighbours. A simple probability model was used for evaluating the likelihood of a pair of neighbours given a pair of candidate matches, in order to investigate the appropriateness of the underlying Bayesian algorithms for resolving matches in the sparse point matching problem. Although the stereo pairs we used were simple and already rectified, the matching algorithms didn’t take advantage of the structural simplicity of the images; therefore any gains made are intrinsic to probability/belief update algorithm and not so much to the probability models on the structure of the point candidates and neighbour relations. The LBP method consistently outperforms the iterative Bayesian method by a margin which increases with the number of points in the images. For example, in Table 3, the correct match percentage goes from a three percent improvement for LBP over the Perwass style method (95.3 and 92.1 percent respectively) in an image pair where there are 50 points and ANN and ANC are 3.9 and 4.0 respectively, to a nine percent improvement when there are 400 points and ANN and ANC are 30.6 and 31.2 respectively. Both algorithms are qualitatively similar in complexity, with similar

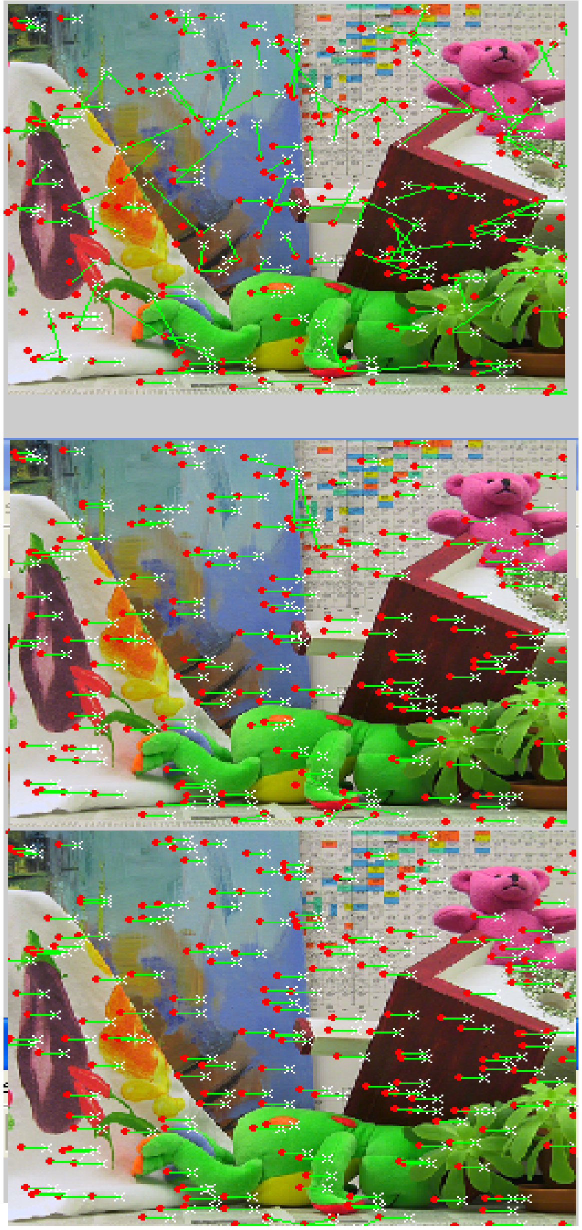

Figure 5: These three frames depict the point correspondences calculated by each algorithm, with 200 points (the search window size was 50 pixels). The top frame shows the unresolved set of correspondences, the middle frame shows the set resolved by the Perwass style iterative Bayesian algorithm, and in the bottom image is shown the results of the LBP algorithm.

run times.

8. Future work

Epipolar geometry could be incorporated probabilistically into this scheme, and a more complicated structure preservation criterion could be imposed. A method for dealing with outliers could easily be included using this framework, for example by allowing an additional entry in the candidate match state vector of each point in the first image to correspond to a ”no match” label.

9. Acknowledgements

The authors express their gratitude for the financial support given by the National Research Foundation of South Africa, and given by Anglo American corporation through the Minerals Processing Research Unit at the University of Cape Town.

10. References

[1] J.Malik A.C.Berg, T.L.Berg. Shape matching and object recognition using low distortion correspondence. *CVPR*, 2005.

[2] G. Sommer C.B.U. Perwass. Dense image point matching through propagation of local constraints.

[3] R. Szeliski D. Scharstein. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[4] S. Lacroix I. Jung. A robust interest points matching algorithm. *In. Proc. ICCV'01*, pages 538–543, 2001.

[5] J. Costiera J. Maciel. A global solution to sparse correspondence problems. *IEEE Trans. on PAMI*, pages 187–199, 2003.

[6] K. Marias M. I. A. Lourakis, A. A. Argyros. A graph-based approach to corner matching using mutual information as a local similarity measure. *ICPR*, 2004.

[7] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[8] R. Cipolla K. Wood P. Smith, D. Sinclair. Effective corner matching. *In. Proc. BMCV'98*, pages 545–556, 1998.

[9] C. Perwass and G. Sommer. An iterative bayesian technique for dense image point matching. In R.P. Würtz and M.Lappe, editors, *Dynamic Perception*, pages 283–288. Workshop of GI section 1.0.4 “Image Understanding” and the European Networks MUHCI and ECOVISION, Ruhr-University Bochum, Nov. 2002, IOS Press infix, 2003.

[10] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. *In. Proc. CVPR'97*, pages 261–266, 1997.

[11] A. Zisserman R. Hartley. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[12] T. Skordas R. Horaud. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans on PAMI*, pages 1168–1180, 1989.

[13] P.H.S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 2002.

[14] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979.

[15] Z. Zhang. A new and efficient iterative approach to image matching. *In. Proc. of ICPR*, pages 563–565, 1994.