IMPROVED SPARSE CORRESPONDENCE RESOLUTION USING LOOPY BELIEF PROPAGATION WITH MRF CLIQUE BASED STRUCTURE PRESERVATION

Markus Louw Fred Nicolls
Department of Electrical Engineering
University of Cape Town
South AFrica
email: markus.louw@gmail.com

ABSTRACT
In this paper we develop a novel MRF formulation for calculating sparse features correspondence in image pairs. Our MRF terms can include cliques of variable sizes, and solve these using Loopy Belief Propagation. To calculate our MRF topology we develop a variant of the K-means algorithm which we call the KN-means algorithm (where each mean has a specified number of neighbours). The method is compared to other state of the art sparse feature correspondence algorithms and shown to compare well, especially for less dense feature sets. Outliers are handled naturally within this paradigm.

KEY WORDS
Feature matching, MRF, LBP, sparse correspondence.

1 Introduction and Literature Review

The goal 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 [12]. 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 usual initial step of doing a local patch correlation to compare each source point to its candidates, and assigning normalized probabilities for each point based on this correlation, usually results in matching incompatibilities (if the MAP candidate is chosen for each source point), and ignores useful information about the relative orientations and neighbourhood structures of the points or features. The algorithm here described could be used for initialization of a dense stereo matcher, for biometric identification, for surface or image (2D or 3D) registration, or even for target tracking.

One of the first descriptions of the characteristics of a good feature matching algorithm was given by Ullman [14], where three principles for matching were given, viz. the principles of *similarity, proximity and 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 [9], 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 [13], maximal cliques in a relational subgraph are established, and in [5] a concave programming approach was used. Iterative closest point algorithms [11], [6] are also feature matching algorithms.

In [7], a simple pairwise MRF based approach was developed and was shown to work well for sparse matching on pre-rectified images, however the structure preservation criterion was too simple and the structure of the clique about any point could only be taken into account indirectly in the MRF potential terms. Also in [7] an iterative Bayesian algorithm based on the dense stereo formulation of [2] was developed. Our algorithm is later compared to both of these algorithms.

In this paper we develop a more general MRF based structure criterion in which the entire local clique structure about local point means within the point set may be taken into account in the energy term. Almost any derived characteristic of the structure may be used (interior angles, distance from centroid, Gaussian curvature) as a term in the energy function of the clique's candidate target points, which is then (approximately) minimized by a Loopy Belief Propagation algorithm.

2 Neighbours and candidates

We are given two point sets X_1 and X_2, in the first and second images respectively. After we run the KN-means algorithm on X_1, we have K mean points. The i_{th} such mean has $n(i)$ neighbours. For each j_{th} point $X_1(j)$ in X_1 we make a list $C(j)$ of candidate points in X_2 for that point. The notation is overloaded in such a way that if $C(j)$ refers to candidate matches in X_2 for point $X_1(j)$, and if there are $n(C(j))$ number of candidate matches we may refer to the k_{th} such point as $C(j,k)$. Similarly we want to refer to the k_{th} neighbour of mean $K_1(i)$ as $N_K(i,k)$. This scheme is depicted in Fig. 1.

To see a real example of feature points and their can-

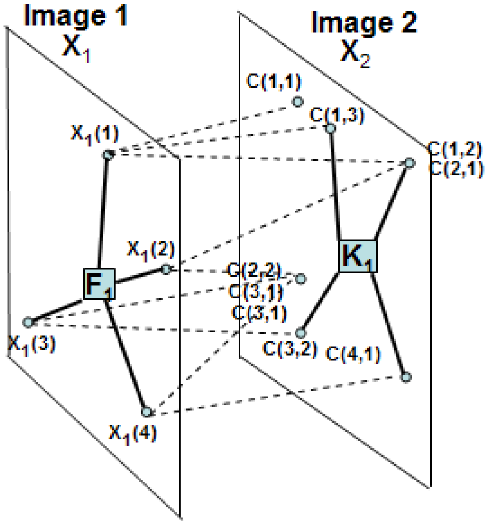

Figure 1. This figure shows the neighbourhood and candidate schemes for X_1 in the first image. The quadrilateral on the left is the first image, containing four points, each labelled as $X_1(i)$ with $i = 1..4$. The points are connected to a mean/factor node (joined by thick lines). Each point's candidates in the second image (the quadrilateral on the right) shown by connectivity with a dotted line.

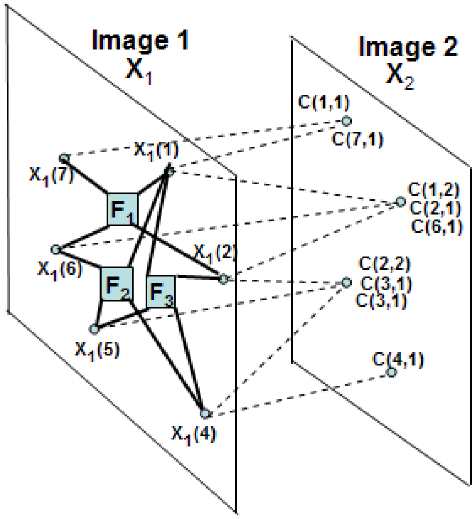

Figure 2. This figure shows that a high number of means (factor nodes) may be connected to the same point/feature set).

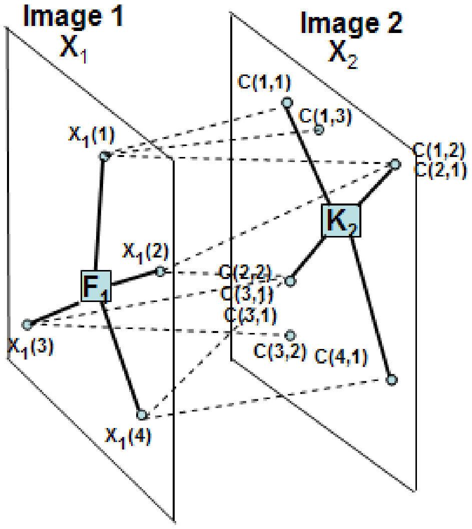

Figure 3. This figure shows that a different set of candidate points for a neighbours of a given factor node will generate a different mean in the target point set (compare location of K1 in Fig. 1).

didate points see Fig. 4, where lines are shown between each source point and its candidate target points. In Fig. 5 we see the means derived from the source point set, and superimposed on the left image of the stereo pair.

3 KN-means algorithm

This method necessitates a means for finding optimal locations for the factor nodes, such that each one has a specified number of neighbours (without duplicate factor nodes with identical connectivity to the same set of neighbours). Furthemore, we should like to specify the number of factor nodes. The algorithm we developed is similar to K-means clustering, and is described as follows:

```
1. Specify $K$ and $N$, the number of means required and
   the number of neighbours for each of these means.
2. Randomly initialize the $K$ means onto non-duplicate
   data points. (If there are more means than data points,
   we can perturb the initial locations with Gaussian
   noise)
3. change $\leftarrow 1$
4. while change$= 1$
5.     change $\leftarrow 0$
6.     for $i = 1..M$
7.         Collect nearest $N$ neighbours for mean $i$
8.         if the neighbours of mean $i$ have changed
```

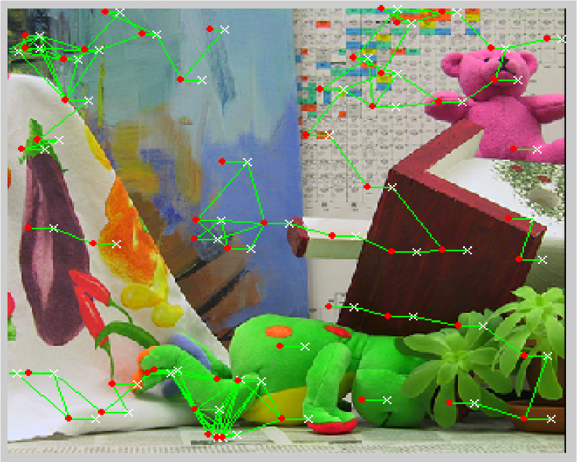

Figure 4. Example of feature points (crosses) and their candidates (dots) superimposed on the first image of the image pair.

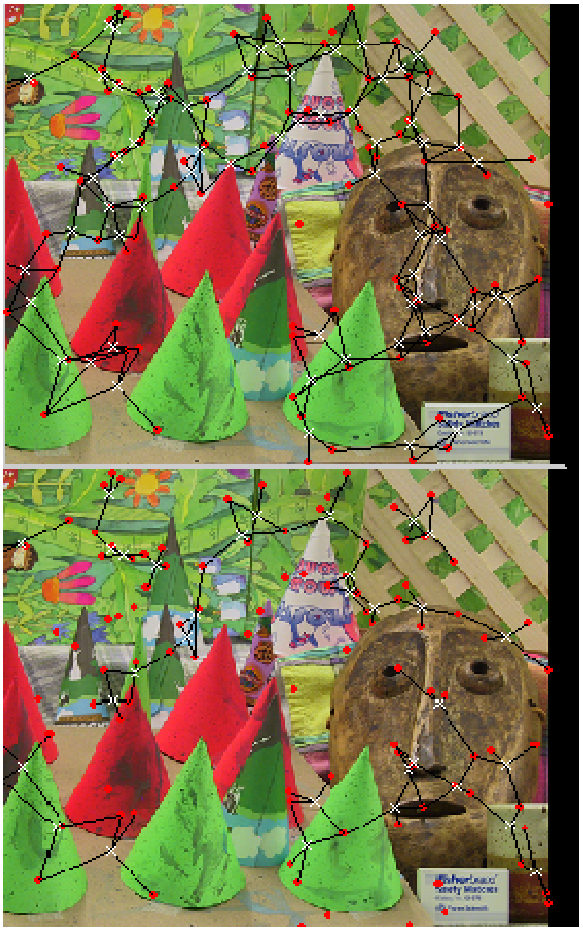

Figure 5. In above images, the feature points are red and the means/factor node locations are shown as white crosses. Green lines connect the factor nodes to each of their neighbours. The lower image shows the image and point set with 25 means, the upper image has 50 means.

```
            Reset $i_{th}$ mean to the average of its
            neighbours
            change$\leftarrow 1$
        end
    end
    Remove duplicate means (means with the same $N$
    neighbours)
end
```

Interestingly, the means which we thus calculate become the factor nodes in the LBP part of the algorithm (each mean becomes a factor node which is connected to the N neighbours of the mean-the location of the mean is not used in the factor node except for calculation of its energy terms). The above algorithm converges reliably enough for our purposes. If a lack of convergence is detected, we try reinitializing the locations of the means (factor nodes), or reducing their number.

4 Resolving sparse correspondences using Loopy Belief Propagation with higher order energy terms

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 $X_1(j)$ to point $C(j,k)$ 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 [8]):

$$\mu_{x\rightarrow f}(x) = o(x)\prod_{g\neq f}\mu_{g\rightarrow x}(x) \tag{1}$$

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

Where x, y are variable nodes, g, f are factor nodes, $\mu(.)$ is a message vector 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. 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 cliques of varying sizes as,

$$p(C|Y) \propto \prod_{k=1}^{P} \exp(-\psi_k(c_k, y_k)) \prod_{i=1}^{K} \exp(-\phi_i(\overrightarrow{c})) \quad (3)$$

where C refers to the correspondence labelling of each point in X_1, P is the number of points to be matched in the first point set, 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, $\psi_k(c_k, y_k)$ 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 2D window correlation matcher), K is the number of factor node energy terms, $\phi_i(\overrightarrow{c})$ is the energy on a particular set of candidate matches for the neighbours of factor node i.

The energy for a point match is given by

$$\psi_k(c_k, y_k) = s\big(X_1(k), C(X_1(k), c_k)\big)\big) \quad (4)$$

where $s(\cdot, \cdot)$ is a matching function between points across images, which returns a normalized probability for the likelihood of the match being correct. This measure only uses relative distances: the squared error distances between each neighbour and the current factor node location are aggregated.

The distance based potential function for a set of candidate labels for a particular factor node is:

$$\phi_i(\overrightarrow{c}) =$$
$$\begin{cases} \sum_{j=1}^{n(i)} \big(|\textbf{dist}\big(X_1(N(i,j), \overrightarrow{c}(j)), K_i\big) - \cdots \\ \quad \textbf{dist}\big(X_2(C(N(i,j)), \overrightarrow{c}(j)), R_i\big)|/\sigma\big) \\ \text{if } \forall kl, \;\; X_2\big(C(N(i,k), \overrightarrow{c}(k)) \neq X_2\big(C(N(i,l), \overrightarrow{c}(l))\big) \\ z \qquad \text{otherwise} \end{cases}$$

In the above equation, the argument vector of $\overrightarrow{c}$ in $\phi_i(\overrightarrow{c})$ is a vector of elements of target indices for each neighbour of the i_{th} factor node. E.g. if $\overrightarrow{c} = [k \;\; l \;\; m \;\; .. \;\; z]$ is a parameter in $\phi_i(\overrightarrow{c})$, this indicates that point $N(i,1)$ is matched to $C(N(i,1),k)$, $N(i,2)$ is matched to $C(N(i,1),l)$, and so on. The energy term for the joint hypothesis over the clique about a factor node depends on the specific candidate match hypothesis for each factor node neighbour; **dist(·,·)** is a measure of the Euclidean distance between the two points indexed by its arguments. Crucially, the mean R_i is a mean which exists in the second image, by taking the centroid of all the candidates (as specified in $\overrightarrow{c}$) of the neighbours of factor node i. R_i must therefore be recalculated for each new set of candidate hypotheses. The z energy term is some small number greater than zero. (We used $z = 0.01$. If $z = 0$, match possibilities are immediately excluded, and it converges incorrectly). After convergence of the LBP algorithm, 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.

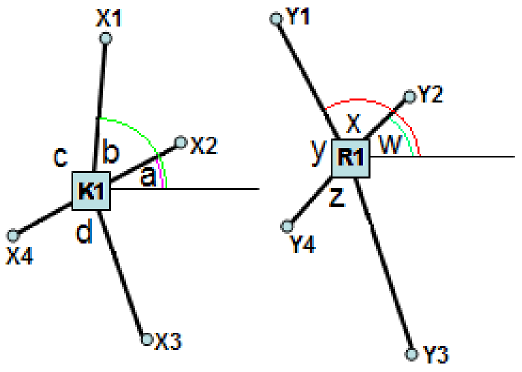

Figure 6. This figure shows the interior angles made by the the Factor node and its neighbours in the first and second images from Fig. 1. The square represents a mean/factor node

To illustrate that we can use almost any structural characteristic using this framework, we show the following potential term based on the angular change per neighbour candidate, as one iterates through the neighbours of a factor node. First define, for the i_{th} factor node

$$\text{next}_i(j) = \quad j+1 \quad \text{if } j+1 < n(i) \quad (5)$$
$$\text{next}_i(j) = \quad 1 \quad \text{if } j+1 = n(i) \quad (6)$$

if $n(i)$ is the number of neighbours for the i_{th}factor node.

$$\phi_i(\overrightarrow{c}) =$$
$$\begin{cases} \sum_{j=1}^{n(i)} \big(|\theta\big(K_i, j, \text{next}(j)\big)\big) - \cdots \\ \quad \theta\big(R_i, C(N(i,j)), C(N(i, \text{next}(j)))\big)\big)|/\sigma\big) \\ \text{if } \forall kl, \;\; X_2\big(C(N(i,k), \overrightarrow{c}(k)) \neq X_2\big(C(N(i,l), \overrightarrow{c}(l))\big) \\ z \qquad \text{otherwise} \end{cases}$$

where in this equation, θ is a function which returns the difference in angles between the positive X axis through the mean and each of the candidate points. This is depicted in Fig. 6, where for example

$$\theta_i(K_1, X_1, X_2) = b - a$$
$$\theta_i(R_1, Y_1, Y_2) = x - w \quad (7)$$

4.1 Outliers

To include support for outliers algorithmically, we can add an element to each vector of candidate probabilities which represents the notion that that point has no corresponding point in the target point set. We therefore modify equation Eqn. to be

$$\phi_i(\overrightarrow{c}) =$$

$$\begin{cases} \sum_{j=1}^{n(i)} \big(|\mathbf{dist}\big(X_1(N(i,j), \overrightarrow{c}(j)), K_i\big) - \cdots \\ \quad \mathbf{dist}\big(X_2(C(N(i,j)), \overrightarrow{c}(j)), R_i\big)|/\sigma\big) \\ \text{if } \forall kl, \;\; X_2\big(C(N(i,k), \overrightarrow{c}(k)) \neq X_2\big(C(N(i,l), \overrightarrow{c}(l))\big) \\ \quad \text{and} \overrightarrow{c_j} \neq \text{ outlier index} \\ y \quad \text{if } \overrightarrow{c_j} = \text{outlier index} \\ z \quad \text{otherwise} \end{cases}$$

In the above equation, y and z are values close to zero.

5 Match measure

In this experiment, we use a modified 2D correlation function which returns values in the range [0,1] for $s(x_A, x_B)$, where a high value indicates a good match. However, our algorithm may be used with a range of different possibilities such as 2D-correlation, Kullbeck-Leibler divergence, Mutual Information, Earth Mover's Distance, etc.

6 Results

Following the methodology of [7], we tested this correspondence resolution algorithm by using ground truth dense stereo pairs from the Middlebury data set, documented in [3]. Random points were chosen from the first image in the stereo pairs, then the ground truth disparity map was used to find the correct corresponding points in the second image. The Middlebury [3] ground truth pairs, which were estimated using a structured lighting approach, provides a useful 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 generated. The scenes of the stereo pairs are natural however, so this algorithm has been tested in "real world" circumstances.

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 is derived. This is seen in the columns for the "Max corr", "LBP" and "Perwass" methods in Tables 1 and 2.

The experimental results obtained from applying these algorithms 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 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 200 runs. In Fig. 7 we see a typical run of KN-LBP on 300 points.

N	Max	KN-LBP	pairwise	Perwass	ANC	ANN
	corr		LBP	sparse		
50	87.6	99.3	87.3	87.1	0.5	0.5
80	81.7	99.6	85.5	85.1	0.85	0.86
100	78.5	99.5	86.1	85.6	1.1	1.1
120	74.8	99.7	86.8	85.8	1.3	1.3
150	71.1	99.6	87.6	86.2	1.6	1.6
200	66.0	96.7	89.3	87.3	2.2	2.2
300	56.9	93.8	91.5	86.9	3.3	3.3
400	50.5	92.5	92.8	67.7	4.3	4.3

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

N	Max	KN-LBP	pairwise	Perwass	ANC	ANN
	corr		LBP	sparse		
50	72.3	99.3	88.8	87.5	2.0	2.1
80	62.5	98.5	91.2	88.9	3.1	3.1
100	58.6	99.5	95.4	91.1	4.1	4.2
120	54.0	99.3	97.4	93.0	4.8	4.8
150	48.7	99.6	97.8	92.5	6.1	6.2
200	44.1	97.5	98.2	92.0	8.2	8.3
300	34.2	92.7	99.4	92.6	12.4	12.4
400	31.7	91.4	99.5	92.8	16.5	16.6

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

N	KN-LBP	pairwise	Perwass
		LBP	sparse
50	0.15	0.001	0.001
80	1.26	0.015	0.012
100	4.7	0.027	0.032
120	9.7	0.038	0.041
150	4.13	0.053	0.057
200	10.8	0.29	0.31
300	89.2	2.42	2.46
400	236.3	11.4	3.12

Table 3. Average time taken in seconds for each algorithm running on different size data sets (candidate target and neighbour size of 30 pixels, 4 neighbours per factor node for KN-LBP), using an Intel Centrino Duo processor

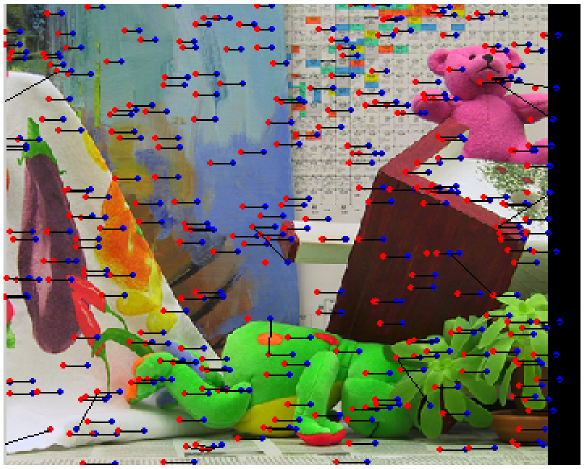

Figure 7. Match correspondence results for 300 source points, 150 Factor nodes, 4 neighbours per factor node. This run shows a 95% correct match resolution.

7 Discussion

Our algorithm performs better than the pairwise LBP methods for point sets below 200. After this, our algorithm begins to scale badly in terms of running time and performance. It is important to the algorithm's performance to choose good values for σ in Eqns. 4.1, 7 and 8. The best value will change depending on things like the density of points and the number of neighbours per factor node.

8 Conclusion

The algorithm we propose performs better than the three alternative methods tested (maximum correlation, and the two iterative Bayesian methods entitled entitled "pairwise LBP" and "Perwass" in the results tables). Although the stereo pairs we used were simple and already rectified, the matching algorithms didn't take advantage of the known epipolar geometry between the images; any gains made are thus intrinsic to the algorithm. The KN-LBP method with clique based structure preservation outperforms the other methods especially when the point sets have a low density (fewer than 200 points). The KN-LBP method requires much more computation, but has much flexibility in terms of taking structural properties into the matching criteria. This algorithm represents an advance in feature matching, a fundamental and general area of computer vision, which is simultaneously addressed by well known approaches like subgraph matching and dynamic programming for a range of applications from mesh registration to feature mapping.

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.

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] M. Louw, F. Nicolls, and D. Bradshaw. An approximate EM homographical iterative closest point algorithm. *In. Proc. Pattern Recognition Association of South Africa*, 2005.

[7] M. Louw, F. Nicolls, and D. Bradshaw. A comparison of iterative bayesian methods for solving the sparse point correspondence problem. *In. Proc. Pattern Recognition Association of South Africa*, 2006.

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

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

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

[11] N.D. McKay P.J. Besl. A method for registration of 3d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1992.

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

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

[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.