

ACCELERATED SPARSE FEATURE 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 extend the work of (Louw and Nicolls, 2007) which proposed a novel Markov Random Field formulation for resolving sparse features correspondences in image pairs. The MRF terms can include cliques of variable sizes, and the energies are minimized using Loopy Belief Propagation. In this paper, an improved MRF topology is developed which uses a variant of the previously developed KN-means algorithm (where each mean has a specified number of neighbours). The message passing schedule is an accelerated one which converges faster than the usual parallel message update schedule, and surprisingly, often gives better correspondence results. The method is compared to other state of the art sparse feature correspondence algorithms and shown to compare well. Outliers are handled naturally within this paradigm.

KEY WORDS

LBP, MRF, sparse feature 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 [3]. 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 ini-

tialization 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 [16], 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 [14], the "some winners take all" of [17], an SVD proximity matrix approximation [12], and resolution by estimation of approximate affine transformations between matches in [5]. In [4], maximal cliques in a relational subgraph are established, and in [9] a concave programming approach was used. Iterative closest point algorithms [2], [7] are also feature matching algorithms.

In [8], 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 [8] an iterative Bayesian algorithm based on the dense stereo formulation of [11] was developed. Our algorithm is later compared to both of these algorithms.

In [6] a more general MRF based structure criterion was developed, 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.

In this paper we show how to improve the KN-means algorithm for this application, how to adjust the parameters for high-order MRF energy terms when calculating the factor nodes, and how to implement an accelerated update schedule for Loopy Belief Propagation using factor nodes, to more rapidly minimize the energy for these higher order MRF energy terms.

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

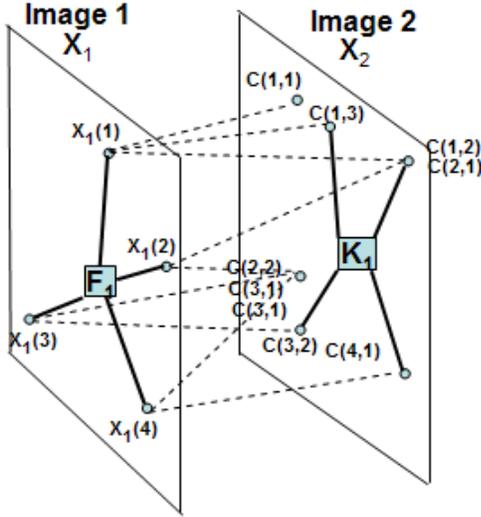


Figure 1. This figure shows the neighbourhood and candidate schemes for X_1 in the first image. The quadrilateral on the left is the first image, containing four points, each labelled as $X_1(i)$ with $i = 1..4$. The points are connected to a mean/factor node (joined by thick lines). Each point's candidates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

points and their candidate points see Fig. 3, where lines are shown between each source point and its candidate target points. In Fig. 4 we see the means derived from the source point set, and superimposed on the left image of the stereo pair.

3 Markov random field potential energy interactions for sparse feature correspondence resolution

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

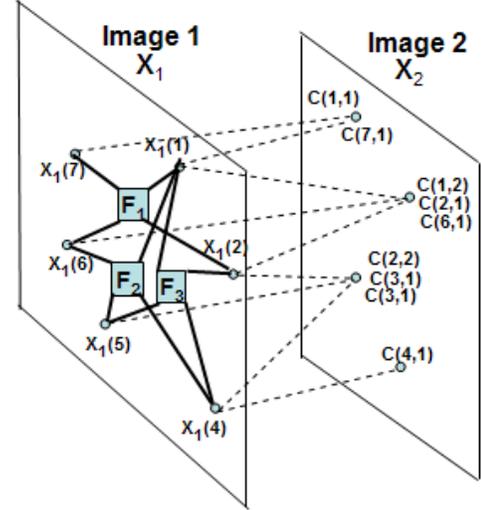


Figure 2. This figure shows that a high number of means (factor nodes) may be connected to the same point/feature set).

sizes as,

$$p(C|Y) \propto \prod_{k=1}^P \exp(-\psi_k(c_k, y_k)/\sigma_k^v) \prod_{i=1}^K \exp(-\phi_i(\vec{c})/\sigma_i^f) \quad (1)$$

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, $\phi_i(\vec{c})$ 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

$$\psi_k(c_k, y_k) = s(X_1(k), C(X_1(k), c_k)) \quad (2)$$

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:

$$\phi_i(\vec{c}) = \begin{cases} \sum_{j=1}^{n(i)} (|\mathbf{dist}(X_1(N(i, j), \vec{c}(j)), K_i) - \dots \\ \mathbf{dist}(X_2(C(N(i, j), \vec{c}(j)), R_i)|/\sigma_i^f) \\ \text{if } \forall kl, X_2(C(N(i, k), \vec{c}(k))) \neq X_2(C(N(i, l), \vec{c}(l))) \\ z \\ \text{otherwise} \end{cases}$$



Figure 3. Example of feature points (crosses) and their candidates (dots) superimposed on the first image of the image pair.

In the above equation, the argument vector of \vec{c} in $\phi_i(\vec{c})$ is a vector of elements of target indices for each neighbour of the i^{th} factor node. E.g. if $\vec{c} = [k \ l \ m \ .. \ z]$ is a parameter in $\phi_i(\vec{c})$, 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; $\text{dist}(\cdot, \cdot)$ is a measure of the Euclidean distance between the two points indexed by its arguments. The mean R_i is a mean which exists in the second image, is calculated by taking the centroid of all the candidates (as specified in \vec{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). We found it useful to check the contents of the factor nodes after calculation, since a bad choice of σ_i^f could cause all the entries to be zero. If this is the case, σ_i^f should be increased until it contains some non-zero values.

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. 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

$$\text{next}_i(j) = j + 1 \quad \text{if } j + 1 < n(i) \quad (3)$$

$$\text{next}_i(j) = 1 \quad \text{if } j + 1 = n(i) \quad (4)$$



Figure 4. In above images, the feature points are red and the means/factor node locations are shown as white crosses. Black lines connect the factor nodes to each of their neighbors. The upper image (reference image) shows the image and point set with 42 means. The points must be matched to their candidates in the target (lower) image

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

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

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. 5, where for example

$$\begin{aligned} \theta_i(K_1, X_1, X_2) &= b - a \\ \theta_i(R_1, Y_1, Y_2) &= x - w \end{aligned} \quad (5)$$

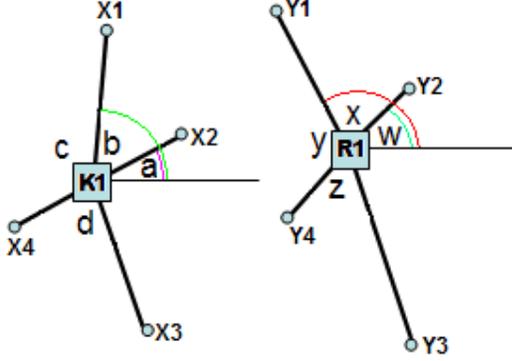


Figure 5. This figure shows the interior angles made by the factor node and its neighbours in the first and second images from Fig. 1. The square represents a mean/factor node

3.1 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

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

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

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 parallel message passing schedule for the Loopy Belief Propagation algorithm using Factor nodes (taken approximately from [10]):

$$\mu_{x \rightarrow f}(x) = o(x) \prod_{g \neq f} \mu_{g \rightarrow x}(x) \quad (6)$$

$$\mu_{f \rightarrow x}(x) = \sum_{u \setminus x} f(u) \prod_{y \neq x} \mu_{y \rightarrow f}(y) \quad (7)$$

Where x, y are variable nodes, g, f are factor nodes, $\mu(\cdot)$ 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 experiments, we used the max-product update algorithm.

4.1 Populating contents of Factor nodes

The factor node entries are populated such that entry \vec{c} for the i^{th} factor node is given a value:

$$F_i(\vec{c}) = \exp(-\phi_i(\vec{c})/\sigma_i^f) \quad (8)$$

where \vec{c} indicates the set of indices in the array, as well as the vector of the current candidate for each of its neighbouring variable nodes (feature points).

5 Stapled KN-means algorithm with removal of duplicates

The MRF topology method we use 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). The algorithm we developed is similar to KN-means clustering algorithm developed in [], with the important difference that each factor node is stapled to a single point feature, i.e. three of its neighbours may change, but the first neighbour is always the same feature point. The number of means is therefore initialized to be the same as the number of point features, but after the duplicate means are removed, there will be fewer of them than there are point features. The algorithm is described as follows:

1. Specify $K = n$, the number of means required, and N , the number of neighbours for each of these means.
2. Initialize the K means onto non-duplicate data points.
3. Set the 1st neighbour of the i^{th} mean to be i , the index of the corresponding feature point.
4. change $\leftarrow 1$
5. while change = 1
6. change $\leftarrow 0$
7. for $i = 1..K$
8. Collect nearest $N - 1$ neighbours for i^{th} mean (adding to this the 1st neighbour, which cannot be removed from this list).
9. If the i^{th} mean's neighbours have changed since the previous iteration
10. Reset i^{th} mean to the average of its N neighbours
11. change $\leftarrow 1$
12. end
13. end
14. Remove duplicate means (means with the same N

15. neighbours)
16. 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.

6 Accelerated message passing schedule for LBP using factor nodes

The accelerated message passing schedule proposed here was inspired by the accelerated message passing schedule due to Tappen [15]. However as it is presented in [15], it is applicable only to LBP without Factor nodes, for the case of pairwise MRF interaction terms only. The method presented here applies to the case of LBP for higher order MRF potential terms, which can only be solved using factor nodes (if belief propagation is used). This method is therefore a generalization of the accelerated schedule of [15], as it may be used to accelerate LBP for an MRF with energy terms of any order. The idea is to pass information through the Bayesian network as quickly as possible. The usual parallel message passing schedule for LBP with Factor nodes iterates between calculating all variable node messages, then all Factor node messages. However, with a graph of span C , it would then take C iterations for information to pass from one side of the graph to the other. The accelerated LBP message passing schedule allows information to be passed more quickly. Once a single factor node message has been calculated, the next node in the visitation list (a variable node) will use this incoming factor node message immediately to calculate its next message (to some factor node). A single visitation list therefore takes the form $\{(p_1, n_1), (f_1, m_1), \dots, (p_N, n_N), (f_N, m_N)\}$, where p_i refers to the index of the i^{th} variable node in the list, and n_i its n^{th} neighbour, and f_i refers to the index of the i^{th} factor node in the list, and m_i its m^{th} neighbour.

6.1 Forming the node visitation lists

In each iteration through all the visitation lists, we want new information to spread as quickly as possible. Therefore we want the visitation lists to be as long as possible, and to have as few lists as possible. However, all messages must be calculated once per iteration through all the visitation lists. Since the variable nodes and the Factor nodes are similar nodes in a graph, we can use a recursive algorithm to traverse the graph repeatedly, forming new lists of nodes to visit. To find such paths in the factor graph, at each node visitation, it is necessary to establish, for each neighbour of

the current node

- If the node has any neighbours where the path to that neighbour is unused over all path traversals thusfar
- If the node has any unvisited neighbours (unvisited in the current path formation)
- If the node has any unvisited neighbours which have other unvisited neighbours.

For any node, all neighbours are tested. The order of preference for adding a node to the visitation list is

1. Neighbours meeting all three criteria (highest preference)
2. Neighbours meeting the first two criteria
3. Neighbours meeting only the first criterion (minimal requirement).

A neighbour must meet the first criterion if it is to be added to the current visitation list. If no neighbours of the current node meet the first criterion, the list is ended, and the calculation of the next visitation list can begin. Then, the list of visited nodes is reset to the empty list, but the list of used paths between nodes is maintained.

After the visitation lists have been formed in this way, the list is doubled: in the update algorithm we move once through each of the visitation lists, then, we move through each of the lists backwards, updating the message indicated by the entry in the visitation list. After this is done, all the messages from factor and variable nodes will have been updated once. The extraction of visitation lists is depicted in Figs. 6 to 9. In Fig 6, a Markov network is shown with variable nodes (round) connected to the corresponding Factor nodes (square). The connectivity of variable nodes to factor nodes is indicated by thick black lines. In Fig. 7, the first visitation list has been formed (the path is indicated by the arrows). In Figs. 8 and 9, the 2nd and 3rd visitation lists are shown. Since subsequent visitation lists may not use edges already used (in either the current or previous visitation lists), these are deleted from the figures as the algorithm progresses.

7 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.

8 Results

Following the methodology of [8], we tested this correspondence resolution algorithm by using ground truth

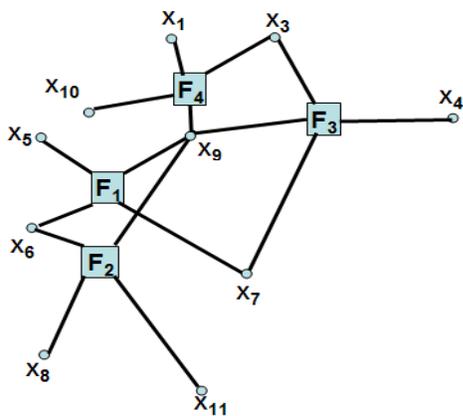


Figure 6. The set of variable nodes (round) connected to factor nodes (square). The thick black lines indicate which variables belong to which energy terms (factor nodes).

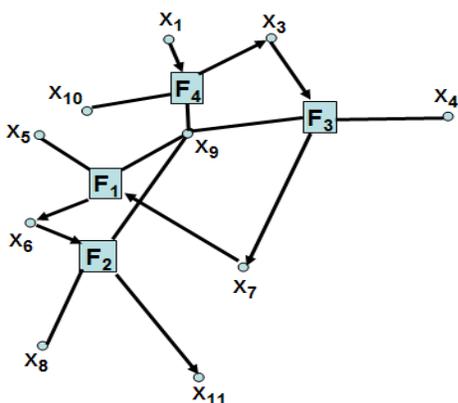


Figure 7. The set of variable nodes (round) connected to factor nodes (square). The thick black lines indicate which variables belong to which energy terms (factor nodes). The thick arrows indicate the first visitation list.

dense stereo pairs from the Middlebury data set, documented in [13]. 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 [13] 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 images were then warped using randomly generated homographies, to make the matching problem more realistic.

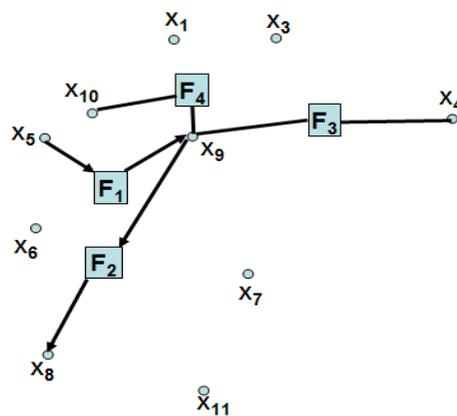


Figure 8. Paths along the previous visitation list are not considered when calculating the second visitation list.

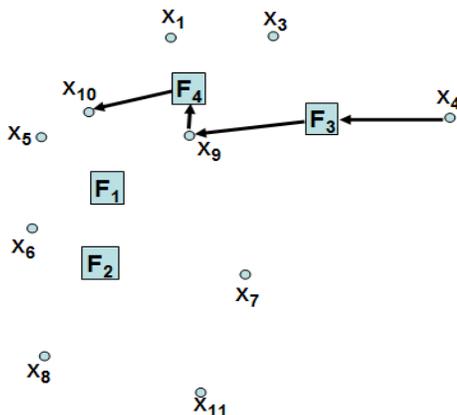


Figure 9. The third visitation list.

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. The procedure is therefore as follows:

1. For $i = 1..100$
2. Generate N points from first image, excluding any point which is occluded in the second view
3. Calculate corresponding points in second image, using disparity map
4. Warp images and points
5. Use algorithm to compute point correspondences
6. end

The match results are seen in the columns for the "Random LBP", "Stapled LBP" and "Random Accel", "Stapled Accel" in Tables 1 to 5, for different numbers of iterations of the algorithm. "Random" refers to there being no restriction on which neighbours a mean can have in the KN-means algorithm, "Stapled" means that the

condition is applied that some feature point must always be a neighbour of any mean. “LBP” refers to the normal parallel message passing schedule, “Accel” refers to the accelerated message passing schedule described in the previous section.

num	Random	Stapled	Random	Stapled	ANC
iters	LBP	LBP	Accel	Accel	
1	33.46	36.86	36.02	39.86	9.16
3	37.29	37.72	39.96	40.56	9.16
5	6.90	37.47	40.21	40.67	9.16
7	37.55	37.66	40.30	40.70	9.16
9	37.66	37.58	40.37	40.70	9.16
11	37.93	37.54	40.37	40.71	9.16

Table 1. Convergence results for four algorithm variations, averaged over 100 runs, with 50 feature points.

num	Random	Stapled	Random	Stapled	ANC
iters	LBP	LBP	Accel	Accel	
1	50.39	55.27	53.51	59.74	10.55
3	55.44	56.87	59.43	60.52	10.55
5	56.32	56.76	59.83	60.62	10.55
7	56.10	56.45	60.01	60.63	10.55
9	56.81	56.78	60.07	60.66	10.55
11	56.80	56.70	60.07	60.68	10.55

Table 2. Convergence results for four algorithm variations, averaged over 100 runs, with 80 feature points.

num	Random	Stapled	Random	Stapled	ANC
iters	LBP	LBP	Accel	Accel	
1	58.80	67.10	63.48	73.74	10.96
3	68.21	69.70	72.68	75.78	10.96
5	68.63	69.17	73.63	75.80	10.96
7	68.72	69.17	73.74	75.78	10.96
9	68.16	69.66	73.82	75.89	10.96
11	67.90	70.06	73.80	75.86	10.96

Table 3. Convergence results for four algorithm variations, averaged over 100 runs, with 100 feature points.

The experimental results obtained from applying these algorithms to point sets of varying sizes are shown in Tables 1 to 5. The search window for finding candidate points for matching in the second image for each point in the first image was held constant at 75 pixels. In these tables, “ANC” is the average number of candidates per point, over 100 runs.

num	Random	Stapled	Random	Stapled	ANC
iters	LBP	LBP	Accel	Accel	
1	71.57	81.23	77.61	89.84	13.73
3	81.50	84.50	88.23	90.76	13.73
5	84.03	84.92	88.80	90.65	13.73
7	83.76	82.53	89.00	90.73	13.73
9	82.26	81.50	89.15	90.73	13.73
11	81.84	83.50	89.19	90.73	13.73

Table 4. Convergence results for four algorithm variations, averaged over 100 runs, with 120 feature points.

num	Random	Stapled	Random	Stapled	ANC
iters	LBP	LBP	Accel	Accel	
1	79.42	95.00	86.42	104.14	15.63
3	95.00	97.57	105.42	107.14	15.63
5	96.42	98.28	108.28	107.14	15.63
7	93.85	96.85	107.42	107.28	15.63
9	99.00	92.85	106.85	108.71	15.63
11	97.00	103.00	106.85	108.71	16.63

Table 5. Convergence results for four algorithm variations, averaged over 100 runs, with 150 feature points.

9 Conclusion

The accelerated message passing algorithm with the stapled KN-means algorithm we propose performs better than the previous version, which was not accelerated and did not use a “stapled” version of the KN-means algorithm. The effects of the stapled algorithm and the accelerated update schedule were separated by disabling one or both of them, as shown in Tables. 1 to 5. These tables also indicate that the accelerated schedule converges faster, and more reliably to the correct MAP MRF configuration. There is also evidence of oscillatory behaviour in the convergence results, which is a side effect of doing belief propagation on loopy networks. Although the stereo pairs we used were simple and already rectified, we made the problem more difficult by warping the images homographically for each trial run. The matching algorithms did not take advantage of the known epipolar geometry between the images; any gains made are thus intrinsic to the algorithm.

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.

num iters	num points	Random LBP	Stapled LBP	Random Accel	Stapled Accel
1	50	0.25	0.28	0.30	0.32
3	50	0.24	0.41	0.33	0.44
5	50	0.25	0.55	0.47	0.56
7	50	0.29	0.54	0.50	0.58
1	100	13.60	18.86	14.42	19.97
3	100	16.74	19.16	18.32	22.67
5	100	20.59	25.73	21.45	28.64
7	100	22.24	27.01	24.86	29.26
1	150	161.92	229.04	159.28	233.39
3	150	185.82	268.03	181.83	285.31
5	150	208.95	330.02	217.82	344.43
7	150	227.46	394.50	244.30	403.43

Table 6. Average running time in seconds per iteration, for each algorithm, run on 50, 100 and 150 feature points, with “ANC” ≈ 10 .

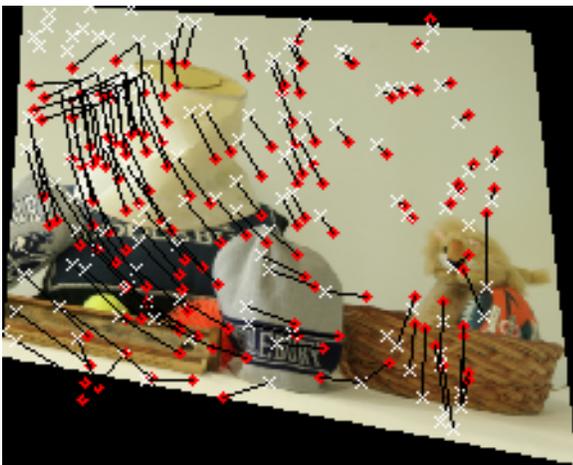


Figure 10. Match correspondence results for 150 feature points, 4 neighbours per factor node. This run shows a 80% correct match resolution.

References

[1] A.C.Berg, T.L.Berg, and J.Malik. Shape Matching and Object Recognition using Low Distortion Correspondence. *CVPR*, 2005.

[2] P.J. Besl and N.D. McKay. A method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1992.

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

[4] R. Horaud and T. Skordas. Stereo Correspondence through Feature Grouping and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1168–1180, 1989.

[5] I. Jung and S. Lacroix. A Robust Interest Points Matching Algorithm. *International Conference on Computer Vision*, pages 538–543, 2001.

[6] M. Louw and F. Nicolls. Improved Sparse Correspondence Resolution Using Loopy Belief Propagation with MRF Clique based Structure Preservation. *In Proc. 7th IASTED Int. Conf. Visualization, Imaging, and Image Processing*, 2007.

[7] M. Louw, F. Nicolls, and D. Bradshaw. An approximate EM Homographical Iterative Closest Point algorithm. *In. Proc. Pattern Recognition Association of South Africa*, 2005.

[8] 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.

[9] J. Maciel and J. Costiera. A Global Solution to Sparse Correspondence Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 187–199, 2003.

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

[11] C.B.U. Perwass and G. Sommer. Dense image point matching through propagation of local constraints. *Bericht Nr. 0205, Christian-Albrechts-Universität Kiel, year=2002*.

[12] M. Pilu. A Direct Method for Stereo Correspondence Based on Singular Value Decomposition. *In. Proc. CVPR*, pages 261–266, 1997.

[13] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 2002.

[14] P. Smith, D. Sinclair, R. Cipolla, and K. Wood. Effective Corner Matching. *In. Proc. BMCV’98*, pages 545–556, 1998.

[15] M. F. Tappen and W. T. Freeman. Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters. *International Conference on Computer Vision*, 2003.

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

[17] Z. Zhang. A New and Efficient Iterative Approach to Image Matching. *In. Proc. of ICPR*, pages 563–565, 1994.