

Efficient Alignment and Correspondence Using Edit Distance

Paolo Bergamini¹, Luigi Cinque¹, Andrew D.J. Cross²,
Edwin R. Hancock^{2*}, Stefano Levialdi¹, and Richard Myers²

¹ Dip. di Scienze dell'Informazione
Università "La Sapienza" di Roma
Via Salaria, 113 - 00198 Rome IT

² Dept. of Computer Science
University of York
Heslington - Y01 5DD York UK

Abstract. This paper presents work aimed at rendering the dual-step EM algorithm of Cross and Hancock more efficient. The original algorithm integrates the processes of point-set alignment and correspondence. The consistency of the pattern of correspondence matches on the Delaunay triangulation of the points is used to gate contributions to the expected log-likelihood function for point-set alignment parameters. However, in its original form the algorithm uses a dictionary of structure-preserving mappings to assess the consistency of match. This proves to be a serious computational bottleneck. In this paper, we show how graph edit-distance can be used to compute the correspondence probabilities more efficiently. In a sensitivity analysis, we show that the edit distance method is not only more efficient, it is also more accurate than the dictionary-based method.

1 Introduction

The matching of point-sets is a problem of central importance in computer vision. The process is usually abstracted as either alignment or correspondence. Alignment is concerned with recovering the set of transformation parameters that bring the points into registration with one-another [10,12]. Correspondence is a symbolic process which is concerned with consistently labelling the points [4,6]. Alignment can be realised using maximum likelihood methods while correspondence is frequently posed as a graph-matching problem.

In the majority of the literature there is a strong dichotomy between the two approaches. However, in a recent paper Cross and Hancock [1] have observed that there are important synergies that can be exploited. Specifically, they have noted that there is a chicken-and-egg problem. Before alignment parameters can be recovered there need to be correspondences available. Correspondence estimation, on the other hand, needs information concerning alignment. In order to overcome this problem, they develop a dual-step EM algorithm [2,5] in which the

* Corresponding author. Email erh@cs.york.ac.uk

consistency of the pattern of correspondences is used to constrain the estimation of alignment parameters. The method is proved effective in the matching of planar point-sets under affine and perspective geometries.

Despite proving effective the method is computationally demanding. The reason for this is that the correspondence probabilities, which weight contributions to the expected log-likelihood function for alignment parameter estimation are computed using a time-consuming dictionary-based method. The aim in this paper is to address this deficiency by using a more efficient method for computing the correspondence probabilities. We turn to the edit-distance model recently reported by Myers, Wilson and Hancock [11]. The main bottle-neck in the Cross and Hancock method [1] is the need to model the effects of structural error by padding the consistent mappings between graphs with dummy nodes. In the edit distance approach, this is simplified by computing the Levenshtein distance between coded strings that represent neighbourhood structure of the graphs being matched. By adopting the edit-distance method for computing the correspondence probabilities, we not only accelerate the Cross and Hancock method, we also increase its accuracy. This is attributable to the fact that the dictionary-based method is more likely to become trapped in local minima.

2 Preliminaries

One of our goals in this paper is to recover the elements $\phi_{k,l}^{(n)}$ which describes a coordinate system transformation that will best bring a model-image feature points set \mathbf{z} into registration with their counterparts in a data set \mathbf{w} . In order to do this, we represent each point in the model set by an augmented position vector $\mathbf{z}_i = (x_i, y_i, 1)^T$ where i is the point index. This augmented vector represents the two-dimensional point position in a homogeneous coordinate system. We will assume that all these points lie on a single plane in the image. In the interests of brevity we will denote the entire set of model points by $\mathbf{z} = \{\mathbf{z}_i, \forall i \in \mathcal{M}\}$ where \mathcal{M} is the points-index set for the model feature-points \mathbf{z}_i . The corresponding fiducial points constituting the data-image are similarly represented by $\mathbf{w} = \{\mathbf{w}_j, \forall j \in \mathcal{D}\}$ where \mathcal{D} denotes the points index-set.

In this paper we are interested in affine transformations, which has six free parameters. These model the two components of translation of the origin on the image plane, the overall rotation of the co-ordinate system, the overall scale, together with the two parameters of shear. These parameters can be combined succinctly into an augmented matrix that takes the form

$$\Phi^{(n)} = \begin{pmatrix} \phi_{1,1}^{(n)} & \phi_{1,2}^{(n)} & \phi_{1,3}^{(n)} \\ \phi_{2,1}^{(n)} & \phi_{2,2}^{(n)} & \phi_{2,3}^{(n)} \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

With this representation, the affine transformation of co-ordinates is computed using the following matrix multiplication

$$\mathbf{z}_i^{(n)} = \Phi^{(n)} \mathbf{z}_i \quad (2)$$

Clearly, this multiplication gives us a vector of the form $\mathbf{z}_i^{(n)} = (x, y, 1)^T$. The superscript n indicates that the parameters are taken from the n^{th} iteration of our algorithm.

The basic idea behind the dual step EM algorithm of Cross and Hancock is to exploit structural constraints to improve the recovery of affine parameters from sets of feature points. Because of its well documented robustness to noise and change of viewpoint, we adopt the Delaunay triangulation as our basic representation of image structure [3]. We establish Delaunay triangulations on the data and the model, by seeding Voronoi tessellations from the feature-points.

The process of Delaunay triangulation generates relational graphs from the two sets of point-features. More formally, the point-sets are the nodes of a data graph $G_D = \{\mathcal{D}, E_D\}$ and a model graph $G_M = \{\mathcal{M}, E_M\}$. Here $E_D \subseteq \mathcal{D} \times \mathcal{D}$ and $E_M \subseteq \mathcal{M} \times \mathcal{M}$ are the edge-sets of the data and model graphs. Key to our matching process is the idea of using the edge-structure of Delaunay graphs to constrain the correspondence matches between the two point-sets. We represent the set of correspondences at iteration n by the function $f^{(n)} : \mathcal{M} \rightarrow \mathcal{D}$. In other words the statement $f^{(n)}(i) = j$ means that the model-point i is in correspondence with data-point j at iteration n of the matching process. In order to construct the expected log-likelihood function we will need to compute the consistency of the arrangement of correspondence matches. We therefore let $\zeta_{i,j}$ denote the probability of the correspondence match $f^{(n)}(i) = j$. In [1], Cross and Hancock observed that the EM algorithm provides a natural framework for recovering the required correspondences and aligned point co-ordinates. The method is concerned with finding maximum likelihood solutions to problems posed in terms of missing or hidden data. According to Cross and Hancock, if the pattern of correspondences $f^{(n)}$ is regarded as missing data, then the task of maximising complete likelihood function $p(\Phi^{(n)}, f^{(n)} | \mathbf{w}, \mathbf{z})$ can be posed as that of maximising the expected log-likelihood

$$Q(\Phi^{(n+1)} | \Phi^{(n)}) = \sum_{j \in \mathcal{D}} \sum_{i \in \mathcal{M}} P(\mathbf{z}_i | \mathbf{w}_j, \Phi^{(n)}) \zeta_{i,j}^{(n)} \ln p(\mathbf{w}_j, \mathbf{z}_i | \Phi^{(n+1)}). \quad (3)$$

Broadly speaking, we can describe the EM algorithm framework as follows. In the Expectation step the *a posteriori* probabilities $P(\mathbf{z}_i | \mathbf{w}_j, \Phi^{(n)})$ of the missing data (i.e. the model-graph measurement vectors, \mathbf{z}_i) are updated by substituting the point positions vector into the conditional measurement distribution. In the Maximization step, by two interleaved substeps the correspondence assignments $f^{(n+1)}(i) = \arg \max_{j \in \mathcal{D}} P(\mathbf{z}_i | \mathbf{w}_j, \Phi^{(n)}) \zeta_{i,j}^{(n)}$ are calculated and the updated alignment matrix

$$\Phi^{(n+1)} = \left[\sum_{(i,j) \in f^{(n)}} P(\mathbf{z}_i | \mathbf{w}_j, \Phi^{(n)}) \zeta_{i,j}^{(n)} \mathbf{z}_i \ B \right]^{-1} \left[\sum_{(i,j) \in f^{(n)}} P(\mathbf{z}_i | \mathbf{w}_j, \Phi^{(n)}) \zeta_{i,j}^{(n)} \mathbf{w}_j \ B \right]$$

is estimated ($B = U^T \mathbf{z}_i^T \Sigma^{-1}$, the elements of the matrix U are the partial derivatives of the affine transformation matrix with respect to the individual parameters and Σ is the variance-covariance matrix for the position errors).

The probabilities $\zeta_{i,j}^{(n)}$ measure the consistency of the pattern of correspondences when the match $f^{(n)}(i) = j$ is made and their computation constitutes the main computational bottleneck. Compared with this, the estimation of transformation parameters represents a relatively small overhead. Our aim in this paper is to find an alternative way to compute $\zeta_{i,j}$ in order to speed up the execution time of the method.

3 The Structural Matching Probabilities

In the original dual-step EM algorithm, the gating probabilities $\zeta_{i,j}^{(n)}$ are computed using a dictionary of structure-preserving mappings between the nodes of the model-graph and the nodes of the data-graph. These structures are subgraphs that consist of neighbourhoods of nodes interconnected by arcs; for convenience we refer to these structural subunits as *supercliques*. The superclique of the node i being matched in the model graph with arc-set E_M is denoted by the set of nodes $C_i = i \cup \{l | (i, l) \in E_M\}$. The matched realisation of this superclique under the mapping function $f^{(n)}$ is $\Gamma_i = \{f^{(n)}(l), \forall l \in C_i\}$. Supercliques are illustrated in panel (a) of figure 1, which shows a graph with two of its super-cliques highlighted.

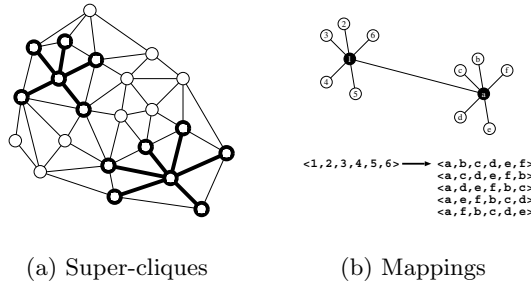


Fig. 1. Supercliques as defined by Wilson and Hancock in [13].

The critical ingredient in developing the matching scheme is the set of feasible mappings between each superclique of the model-graph and those of the data graph. The set of feasible mappings, or dictionary, for the superclique C_i , is denoted by $\Theta_{i,j} = \{S_j\}$ where $S_j = j \cup \{l | (j, l) \in E_D\}$. Each element S_j of $\Theta_{i,j}$, is therefore a relation formed on the nodes of the data-graph; hence, the dictionary of feasible mappings for the superclique C_i consists of all the consistent relations that may be elicited from the data-graph. In practice the dictionary is compiled by considering the cyclic permutations of the non-centre nodes in the superclique C_j about the centre, as shown in panel (b) of figure 1. A complication arises from the fact that not all supercliques have the same size. In [13], Wilson and Hancock addressed this problem by padding the dictionary items with dummy labels so

that it was the same size of the local configuration. This is essentially a brute force method, and may significantly add to the complexity of the dictionaries as we show later.

4 Computing Correspondence Probabilities Using Edit Distance

Recently Myers, Wilson and Hancock have overcome this problem of dictionary padding by showing how $\zeta_{i,j}^{(n)}$ can be computed more efficiently using edit-distance [11]. The Levenshtein or string-edit distance is a measure of the distance between lists of differing lengths [7,8,9]. This avoids the use of padding altogether, by considering insertions and deletions in addition to changes. In what follows, we work with a simplified dictionary $\Theta_{i,j}^c$ which contains only cyclic permutations and whose size is therefore equal to $|C_j| - 1$.

Suppose that $P_{\Gamma_i,S}^*$ is the optimal edit path between the relational image Γ_i and the unpadded dictionary item S . In their recent paper, they have shown that

$$\zeta_{i,j} = \frac{\sum_{S \in \Theta_{i,j}} \exp \left[- \left(k_W W(P_{\Gamma_i,S}^*) + k_L L(P_{\Gamma_i,S}^*) \right) \right]}{\sum_{j \in \mathcal{M}} \sum_{S \in \Theta_{i,j}} \exp \left[- \left(k_W W(P_{\Gamma_i,S}^*) + k_L L(P_{\Gamma_i,S}^*) \right) \right]} \quad (4)$$

where the contents $k_W = \ln \frac{(1-P_e)}{P_e}$ and $k_L = \ln \frac{1}{(1-P_e)}$ are defined in terms of an error-probability P_e .

5 Complexity

As shown by Cross and Hancock in [1] the computation of $\zeta_{i,j}$ is based on the number of dictionary comparisons. For a single step we have that the time complexity is $O(|\Theta_{i,j}| \cdot |S|)$. Moreover, the length of the structure-preserving mappings, $|S|$ is linear in the superclique size and will not play a significant rôle in the overall complexity. In [11], Myers, Wilson and Hancock have formalised that $|\Theta_{i,j}|_{max} = |V_D| \cdot O\left(4^{|C_j|}\right)$ is an upper bound for the total size of the dictionary in terms of the average data graph superclique size $\overline{|C_j|}$ when padding is required. To compute the edit distance between the two strings Γ_i and S we have adopted the standard algorithm described by Wagner and Fischer in [7]. The complexity for a single comparison is $O(|\Gamma_i| \cdot |S|)$. Since $|\Gamma_i| = |C_i|$, $|S| = |C_j|$ and it is sufficient to consider only $|C_j| - 1$ cyclic permutations of C_j , the total size of the dictionary is

$$\begin{aligned} |\Theta_{i,j}| &= \sum_{j \in D} (|C_j| - 1) \cdot O(|C_i||C_j|) \\ &= |V_D| \cdot O(|C_i||C_j|^2) \end{aligned} \quad (5)$$

which is polynomial in the size of the supercliques. Moreover, for Delaunay graphs, the average node degree (and hence the average superclique size) is 6; using this upper bound we can write

$$(|\Theta_{i,j}|_{max})_{[Padding]} = O(4^6) \gg O(6^3) = (|\Theta_{i,j}|)_{[Edit]} \quad (6)$$

Hence, although the edit distance calculation is less efficient than a linear comparison with a padded dictionary, the number of dictionary elements to compare will be much less than with Wilson and Hancock’s padded dictionary approach with a considerable reduction of the execution time.

6 Experiments

In this section, we provide experimental evaluation of our coupled matching process. This investigation has two distinct strands. We present both an algorithm sensitivity analysis and an application on real world imagery.

6.1 Sensitivity Analysis

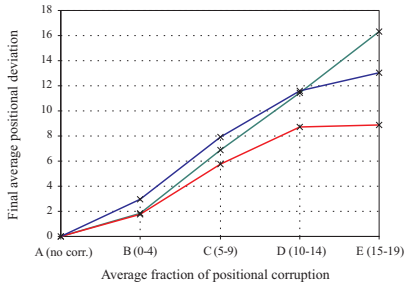
Here we experiment with the edit-distance based dual-step matching scheme and compare it with its dictionary-based counterpart. Our experiments explore the sensitivity of the method to structural corruption and affine rotations. The experiments are based on synthetic graphs generated by randomly distributing 20 nodes in a 200×200 pixel window.

Our sensitivity analysis compares the accuracy of the original padded dictionary based approach with the new edit distance-based method. For both algorithms the initial affine alignment matrix was the identity matrix and the initial correspondences among feature points were all incorrect. To limit the amount of computation required by the original algorithm, the maximum allowed amount of dictionary padding was 2 nodes per superclique. In each of the following plots the red curve indicates the performance of the new edit-distance method, while the blue curve are the results obtained with for Cross and Hancock’s original algorithm. In each figure the left hand panel shows the final alignment error as a function of the average positional deviation and it is expressed in pixels. Suppose that $T \subset \mathcal{M} \times \mathcal{D}$ is the set of ground-truth correspondences between the uncorrupted portion of the data-graph and the model. If n_e is the final iteration number for the matching algorithm, then the measure of registration accuracy is $A = \frac{1}{|T|} \sum_{(i,j) \in T} \left| \mathbf{z}_i^{(n_e)} - \mathbf{w}_j \right|$.

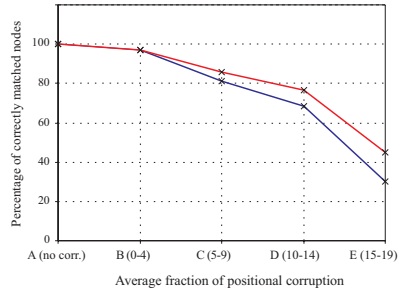
The right-hand panel shows the final fraction of correct correspondences.

We have subjected the data point-set to two types of error. Firstly, we have added varying amounts of Gaussian error to the positions of the points. Here the aim is to simulate the effect of point measurement or localisation error. The second type of error simulates the effects of a poor feature detection. This structural error has been generated randomly adding and deleting points in the data graph and subsequently re-triangulating the point-set.

In Figure 2 we show the effect of varying the amount of Gaussian measurement error. In panel (a) the green curve is the initial registration error. The edit-distance method consistently outperforms the padded dictionary method. When the initial registration error is small, the padded dictionary can actually lead to a deterioration in the alignment.

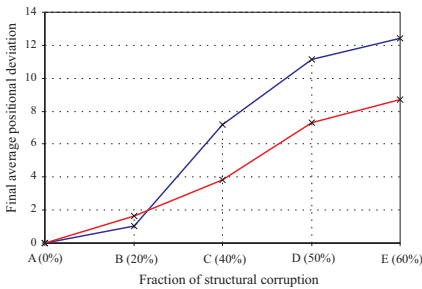


(a) Registration accuracy

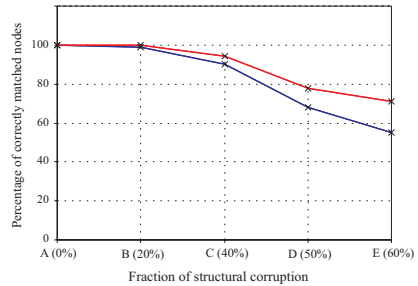


(b) Structural matching results

Fig. 2. Effect of positional Gaussian noise.



(a) Registration accuracy



(b) Structural matching results

Fig. 3. Effect of relational disruption in the data graph.

In Figure 3 we investigate the effect of structural error. Here, both panels show that the new approach is less sensitive to structural corruption. This is because the edit distance based method tolerates any size difference between matching supercliques, whereas the performance of the padded dictionary method rapidly decrease with increasing size difference.

Next we turn our attention to the effects of affine distortion of the point-sets. Here we measure the effects of rotation and Figure 4 we show the obtained results. Here, we have tested the effectiveness of the two methods by progressively rotating the data image with the respect of its geometric centre. The operating limit for the old method has been estimated as $\pm 25^\circ$ while for the edit distance-

based method it is $\pm 35^\circ$. Panel (a) again shows that the new method is less sensitive to noise than the original one.

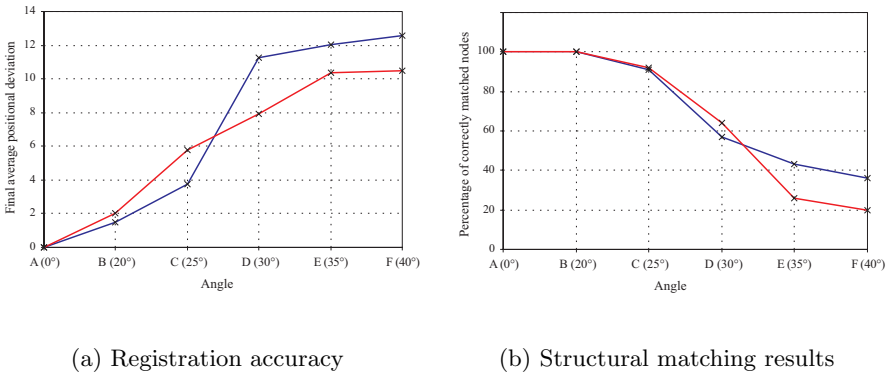


Fig. 4. Sensitivity to affine rotation.

Taken together, these results would suggest that the edit-distance method is not only faster than the padded dictionary method, it is also more accurate. It must be stressed however, that the original method of Cross and Hancock employs two additional refinements not used here. Firstly, it uses an edit process to remove poorly matching nodes. Secondly, it anneals the constant P_e with iteration number. This will reduce problems associated with convergence to a local optimum.

6.2 Real World Imagery

This example demonstrates the effectiveness of the dual-step matching scheme on real world images. We simulate the task of recognizing planar objects in different 2D poses, which is posed by two different images of a 3.5-inch floppy disk.

Panel (a) of Figure 5 shows the model image on which we have superimposed the feature points (corners extracted by hand) and the corresponding Delaunay triangulation. As we can see from panel (b), in which the data object is represented, our experimentation involves at the same time two components of affine transformations: *skewing* and *scaling*.

The sequence in Figure 6 shows the iterative recovery of the affine geometry. Here, we illustrate the iterative registration of the model object against the data image. As in the sensitivity analysis, the initial affine alignment matrix was the identity matrix and the initial correspondences were all incorrect. The first panel of Figure 6 shows the initial situation in the registration process. Each figure in the sequence has been obtained by superimposing the successive model image transformations on the data image. The last panel shows that the process

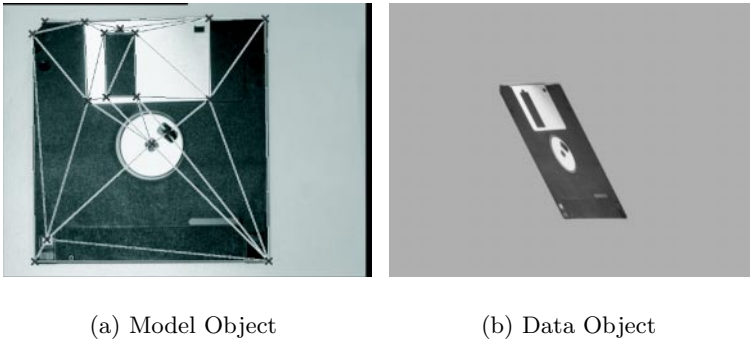


Fig. 5. The two different views used in the matching experiments.

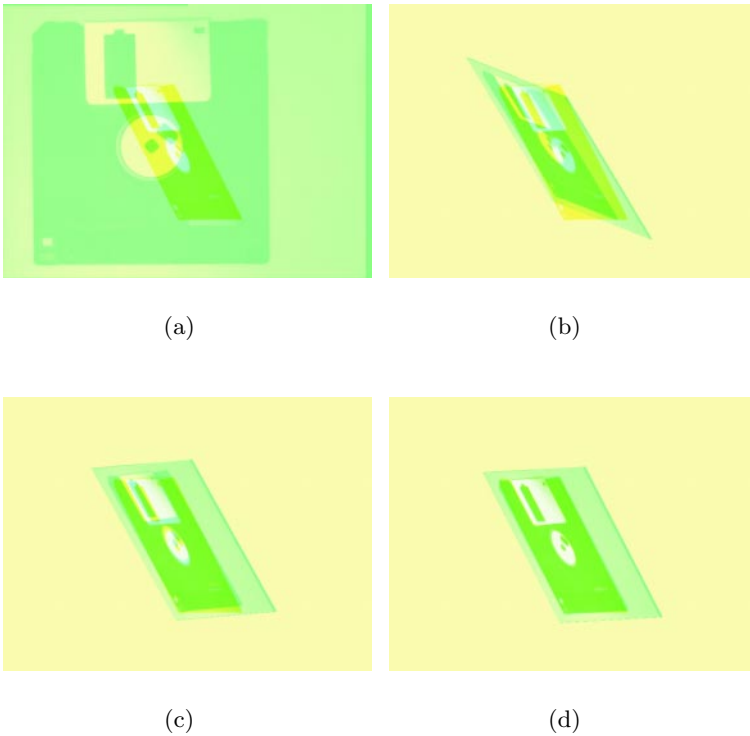


Fig. 6. The iterative registration of the model; the steps are ordered from left-to-right and top-to-bottom.

converges after few iterations; it is clear that the recovered transformation is very accurate even if the initial conditions were unfavourable.

7 Conclusions

In this paper we have shown that the use of edit-distance can improve both the efficiency and accuracy of point-set matching using the Cross and Hancock dual step EM algorithm. Based on these promising results, we intend to extend our work by incorporating several refinements reported in the original work of Cross and Hancock. These include annealing and graph-editing.

References

1. A.D.J.Cross and E.R.Hancock, "Graph Matching with a Dual Step EM Algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, pp 1236-1253, 1998.
2. A.P. Dempster, Laird N.M. and Rubin D.B., "Maximum-likelihood from incomplete data via the EM algorithm", *J. Royal Statistical Soc. Ser. B (methodological)*, **39**, pp 1-38, 1977.
3. O.D. Faugeras, E. Le Bras-Mehlman and J-D. Boissonnat, "Representing Stereo Data with the Delaunay Triangulation", *Artificial Intelligence*, **44**, pp. 41-87, 1990.
4. Gold S., Rangarajan A. and Mjolsness E., "Learning with pre-knowledge: Clustering with point and graph-matching distance measures", *Neural Computation*, **8**, pp. 787-804, 1996.
5. M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, **6**, pp. 181-214, 1994.
6. M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Maalsburg, R.P. Wurtz and W.Konen, "Distortion-invariant object-recognition in a dynamic link architecture", *IEEE Transactions on Computers*, **42**, pp. 300-311, 1993
7. R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem", *Journal of the Association for Computing Machinery*, **21**, pp. 168-173, 1974.
8. A. Marzal and E. Vidal, "Computation of Normalised Edit Distance and Applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, pp. 926-932, 1993.
9. H. Bunke, and J. Csirik, "Parametric String Edit Distance and its application to Pattern Recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, **25**, pp. 202-206, 1995.
10. D.P. McReynolds and D.G. Lowe, "Rigidity Checking of 3D Point Correspondences under Perspective Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, pp. 1174-1185, 1996.
11. R. Myers, R.C. Wilson and E.R. Hancock, "Efficient Relational Matching with Local Edit Distance", *Proceedings of the Fourteenth International Conference on Pattern Recognition*, IEEE Computer Society Press, pp 1711-1714, 1998.
12. D. Oberkampf, D.F. DeMenthon and L.S. Davis, "Iterative Pose Estimation using Coplanar Feature Points", *Computer Vision and Image Understanding*, **63**, pp. 495-511, 1996.
13. R.C. Wilson and E.R. Hancock, "Structural Matching by Discrete Relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, pp. 634-648, 1997.