An Auction Algorithm for Graph-Based Contextual Correspondence Matching

Barend J. van Wyk, Michaël A. van Wyk, and Guillaume Noel

French South-African Technical Institute in Electronics
at the Tshwane University of Technology
Staatsartillerie Road, Pretoria, South Africa
{ben.van.wyk,guillaume.noel,mavw}@fsatie.ac.za

Abstract. The *Auction Graph Matching* (AUGM) algorithm is presented. This algorithm is based on a novel joint probabilistic framework that transforms the graph matching problem into a linear assignment problem which is efficiently solved by the Bertsekas auction algorithm. A salient feature of this single-pass auction-based approach is that the inferred match probabilities are not only constrained over all objects in the reference image, but are also constrained over all objects in the input image.

1 Introduction

Representing the structural descriptions of objects by weighted graphs, reduces the problem of contextual correspondence matching to finding error-correcting graph or sub-graph isomorphisms, also referred to as the *Graph Matching* (GM). As pointed out in Refs. [1], [2] and [3] the graph matching problem is closely related to the *Quadratic Assignment Problem* which can be solved using a variety of neural, annealing, graduated assignment and other iterative methods. In general the *Quadratic Assignment Problem* is more difficult to solve than the *Linear Assignment Problem.* Several breakthroughs to efficiently solve the *Linear Assignment Problem* such as those detailed in Refs. [4] and [5] indicate that there is much to gain if the GM problem can be transformed into a *Linear Assignment Problem.* A method for achieving this transformation is presented in this paper.

Although *Quadratic Assignment* approaches are not in general directly interpretable using Bayesian frameworks such as those detailed in Refs. [6]–[15], the Auction Graph Matching (AUGM) algorithm is built on a Bayesian foundation. A *Joint Probabilistic Framework* is derived in the sequel that leads to a *Linear Assignment Problem* which can be efficiently solved by the Bertsekas auction algorithm [4]. The *Joint Probabilistic Framework* differs significantly from previous probabilistic approaches for contextual correspondence and graph matching, detailed in Refs. [6]–[15], in the following aspects: *(1)* Instead of directly inferring $P(\theta_i = \overline{\theta}_k)$, the probability of associating a vertex in the input graph with a vertex in the reference graph, our main focus is on the inference

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 334–342, 2004.

of the joint probabilities $P(\theta_i = \overline{\theta}_{k,}\theta_j = \overline{\theta}_l)$ from which $P(\theta_i = \overline{\theta}_k)$ are inferred. *(2)* Similar to the work of Kittler *et al.* [8], the AUGM is a single-pass technique. *(3)* Conventional probabilistic methods only constrain the probability of a vertex in the input graph being associated with a vertex in the reference graph over all the vertices in the reference graph. To further minimize the possibility of false matches the AUGM algorithm *also* constrains the probability of a vertex in the input graph being associated with a vertex in the reference graph, over all the vertices in the *input* graph. *(4)* The AUGM algorithm does not rely on the use of compatibility functions specified in terms of the face-units of the graph under match, sparse graph structures or Bayesian edit distances based on these notions.

2 Notation

Suppose an input graphs has n vertices represented by

$$\Omega = \{\theta_1, ..., \theta_n\}.$$

The objective is to calculate the probability of a vertex θ_i in the input graph, being associated with a vertex $\overline{\theta}_k$ in a reference graph having $\bar{n}$ vertices, represented by

$$\overline{\Omega} = \left\{\overline{\theta}_1, ..., \overline{\theta}_{\bar{n}}\right\}.$$

In our framework it is assumed that $\bar{n} \geq n$ and that the probability values $P(\theta_i = \overline{\theta}_k)$ are constraint by

$$\sum_k P(\theta_i = \overline{\theta}_k) = 1, \tag{1}$$

and

$$\sum_i P(\theta_i = \overline{\theta}_k) = 1, \qquad n = \bar{n} \tag{2}$$

$$0 \leq \sum_i P(\theta_i = \overline{\theta}_k) \leq 1, \qquad \bar{n} \geq n. \tag{3}$$

If, in addition it is required that

$$P(\theta_i = \overline{\theta}_k) \in \{0; 1\} \tag{4}$$

then equations 1 to 4 represent the enforcement of two way assignment constraints. Previous Bayesian frameworks were in general not able to enforce constraint 2 or 3.

For each pair of vertices θ_i and θ_j, $i \neq j$, we assume there are s binary measurements corresponding to the attributes of the edges of the input graph:

$$A_{ij} = \left\{A_{ij}^{(1)}, ..., A_{ij}^{(s)}\right\}, \qquad i \neq j, \quad i, j = 1, ..., n.$$

3 Bayesian Reasoning Framework

As usual we will assume that the conditional probability density function

$$p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \tag{5}$$

corresponds to the compatibility coefficients calculated between edges of the input and references graphs using *edge* attributes. See for example Kittler *et al.* [8] , Christmas *et al.* [6] or Faugeras and Price [16]. We will now investigate how Eq. 5 relates to the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$. Our approach drastically differs from previous approaches in the sense that we do not try to estimate or calculate the probabilities $P(\theta_i = \overline{\theta}_k|A_{ij})$ directly. Instead our main computational focus is on the inference of the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ from which all $P(\theta_i = \overline{\theta}_k)$ are estimated using a simple weighted summation process. It is important to note that to infer $P(\theta_i = \overline{\theta}_{k,})$, the probabilities of associating a vertex in the input graph with a vertex in the reference graph, our framework *only relies on edge attributes*. Self edges (self arcs) and vertex attributes are not considered. As a consequence $p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ and $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ where $i = j$ or $k = l$ are not considered in our framework and the independence assumption, i.e. $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_k)P(\theta_j = \overline{\theta}_l)$, holds. Observe that according to Bayes' theorem

$$P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l|A_{ij}) = \frac{p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{\sum_{k,l} p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}, \tag{6}$$

where $i \neq j$ or $k \neq l$. Since all $p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ are fixed (via some compatibility calculation) the only way to maximize the *a posteriori probability* $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l|A_{ij})$ is by adjusting the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$, i.e.

$$P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l|A_{ij}) = \max_{P(\theta_i = \bar{\theta}_k, \theta_j = \bar{\theta}_l)} \left[\frac{p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{\sum_{k,l} p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}\right] \tag{7}$$

where $i \neq j$ or $k \neq l$ subject to the constraints associated with $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ given by

$$\sum_{k,l,i \neq j,k \neq l} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = 1, \tag{8}$$

$$0 \leq \sum_{i,j,i \neq j,k \neq l} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \leq 1, \tag{9}$$

$$0 \leq P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \leq 1. \tag{10}$$

Constraints 8 to 10 were derived using Eqs. 1 to 3 and the fact that $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_k)P(\theta_j = \overline{\theta}_l)$. Finding the constraint joint probabilities

$P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ which maximize $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l | A_{ij})$ over all i, j can be formulated as the following constrained optimization problem

$$\min \sum_{i,j,k,l} \left(p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) - P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)\right)^2 \tag{11}$$

with respect to $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ where $i \neq j$ or $k \neq l$, subject to Eqs. 8 to 10.

3.1 Joint Probability Inference

Since $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_l, \theta_j = \overline{\theta}_k)$ when our graphs are undirected and both these joint probabilities will belong to the same row of an assignment matrix we cannot directly impose the constraint $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \in \{0; 1\}$. However, the fact that $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_l, \theta_j = \overline{\theta}_k)$ when our graphs are undirected implies that not all the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ need to enter Eq. 11, conveniently reducing the dimension of the problem. In fact we only need to consider the indices in the set

$$\{i, j, k, l\}_{j=i+1,...,n \quad l=k+1,...,\bar{n}} \tag{12}$$

where $i = 1, ..., n$ and $k = 1, ..., \bar{n}$ provided that all final values for $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ are halved after solving the lower dimensional problem. It also implies that we can now impose the binary constraint $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \in \{0; 1\}$ on the lower dimensional problem and efficiently solve it using an optimal assignment algorithm. Figure 1 details the time, averaged over a 1000 runs, to execute the Bertsekas auction optimal assignment algorithm [4] written in C on a Pentium IV platform running Windows XP. The generation time of random input matrices is included in our time calculations. As expected the time required per assignment calculation is dependent on the number of undirected edges in the input and reference graphs. For undirected fully-connected graphs of the type considered in this paper the number edges are given by $\eta = \sum_{i=1}^{n-1}(n-i)$ where n represents the number of vertices in the graphs.

Note that the cardinality of the subset $\{i, j\}_{i,j=1,...,n}$ is $\eta = \sum_{i=1}^{n-1}(n-i)$ and that the cardinality of the subset $\{k, l\}_{k,l=1,...,\overline{n}}$ is $\overline{\eta} = \sum_{k=1}^{\overline{n}-1}(\overline{n}-k)$. The dimension of the matrix passed to the Auction algorithm will therefore be $\eta \times \overline{\eta}$. Consequently the rows of the assignment matrix are indexed by $\alpha = 1, ..., \eta$ where α is related to indices i and j by $\alpha = (j-i) + \sum_{s=1}^{i} |U_s|$where $j < i$, U_s is defined as the set $\{i = s, j\}_{j=i+1,...,n}$ and $|\cdot|$ denotes cardinality. Similarly the columns of the assignment matrix are indexed by $\beta = 1, ..., \overline{n}$ where β is related to indices k and l by $\beta = (l-k) + \sum_{s=1}^{k} |U_s|$ where $l < k$ and U_s is defined as the set $\{k = s, l\}_{l=i+1,...,\overline{n}}$.

4 Auction Assignment Algorithm for Graph Matching

Similar to the auction algorithm detailed in Ref. [4] we associate with each column of our assignment matrix (joint probability matrix) a price, say ρ_β. For our application we started by setting all the prices to zero. Let $p_{\alpha\beta} := p(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l | A_{ij})$ where α and β are related to indices i, j, k, l as described in the previous section. A row α is defined as almost happy with a column β assigned to it if

$$p_{\alpha\beta_\alpha} - \rho_{\beta_\alpha} \geq \max_{\beta=1,\ldots,\overline{n}} \{p_{\alpha\beta} - \rho_\beta\} - \delta,$$

where δ is a slack variable.

The auction process then proceeds by selecting a row, α, which has not been assigned a column β_α or is not almost happy. This row finds a column β_α which offers maximal value, i.e.

$$\beta_\alpha \in \arg \max_{\beta=1,\ldots,\overline{n}} \{p_{\alpha\beta} - \rho_\beta\}.$$

Then

1. Row α exchanges its previous β_α (if it had one) with the row its new β_α was assigned to at the beginning of the round.
2. The price ρ_β associated with β_α is set to $\rho_\beta = \rho_\beta + \max_\beta \{p_{\alpha\beta} - \rho_\beta\} - \max_{\beta \neq \beta_\alpha} \{p_{\alpha\beta} - \rho_\beta\} + \delta$.

The process is repeated in a sequence of rounds until all rows α are almost happy. The *slack* value, δ determines how fast the algorithm will converge and how optimal the final answer will be. For all our experiments we have set δ to a small fixed value that will guarantee convergence to an optimal solution as described in [4]. When the procedure terminates the joint probability values

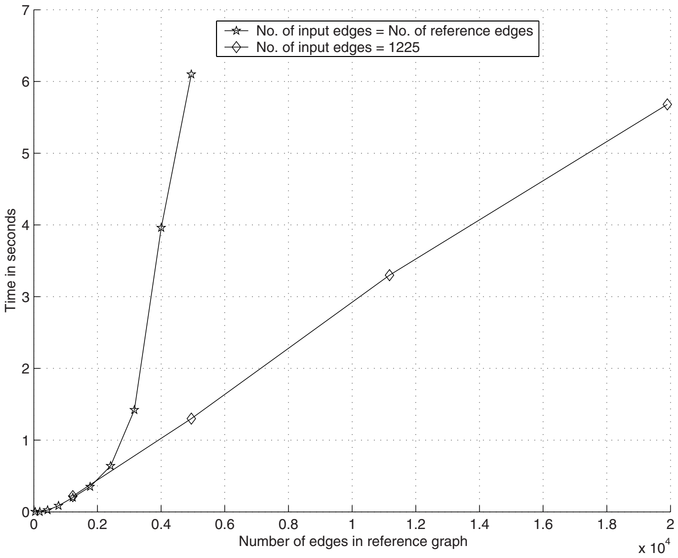

Fig. 1. Execution time of the Bertsekas auction routine.

$P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_j = \overline{\theta}_l, \theta_i = \overline{\theta}_k)$ corresponding to a row-column assignment are set to 0.5 (see previous section for reason why not set to one). The rest of the joint probability values are set to zero.

4.1 Marginal Probability Inference

After obtaining the constraint joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ the desired probabilities $P(\theta_i = \overline{\theta}_k)$ can be inferred using the following proposition:

Proposition 1. *The probabilities inferred by*

$$P(\theta_i = \overline{\theta}_k) = \frac{\sum_{j,l,j \neq i,l \neq k} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{n-1} \tag{13}$$

will satisfy the constraints given by Eqs. 1 to 3 if the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ satisfy the constraints given by Eqs. 8 to 10.

Proof: Observe that $\sum_{k,l,i \neq j,k \neq l} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = 1$ and that $\sum_{j,k,l,j \neq i,k \neq l} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = n - 1$ which implies that $\sum_k \frac{\sum_{j,l,j \neq i,l \neq k} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{n-1} = 1$. Eq. 1 is therefore satisfied. Similarly $0 \leq \sum_{i,j,k \neq l,i \neq j} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \leq 1$ and $\sum_{l,i,j,l \neq k,i \neq j} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \leq \bar{n} - 1$ which implies that $0 \leq \sum_i \frac{\sum_{j,l,j \neq i,l \neq k} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{\bar{n}-1} \leq 1$. Eq. 3 is therefore satisfied since it is assumed that $\bar{n} \geq n$.

4.2 Final Assignment

Once all $P(\theta_i = \overline{\theta}_k)$ are inferred, the most appropriate $\overline{\theta}_k$ for a given θ_i is obtained by

$$\max_{\overline{\theta}_k} \{P\left(\theta_i = \overline{\theta}_k\right)\}_{k=1,\ldots,\bar{n}}. \tag{14}$$

It's easy to see that if all edge attributes are unique and no additive noise is present in the input graph that $\mathbf{P} := \left(P\left(\theta_i = \overline{\theta}_k\right)\right)$ will be an assignment matrix satisfying constraints 1 to 4.

5 Simulation Results

To test the performance of the AUGM algorithm dynamic random line matching experiments, similar to those proposed by Caelli and Caetano [18], were conducted. We prefer this type of experiment above the usual static applications such as stereo line matching, the matching of road networks or buildings as this allow us, through the addition of progressive noise and random line selection, to derive representative (application specific) performance curves for the algorithms included in our comparison. Although lines were used to represent the vertices of a graph, vertex features were not derived nor used. Binary relationships between lines were used as edge attributes. The differences in orientation between lines, line length ratios and distances between line midpoints were used. Refer to Li [19] for more information on the derivation of translation and orientation invariant line features.

For our experiments reference graphs having 50 vertices (derived from a randomly generated line image with line lengths uniformly distributed between 10 and 300 pixels) were used. For the first experiment input graphs were constructed by randomly rotating and translating all 50 lines of the reference line images. To test the robustness of the algorithms against line endpoint anomalies, uniform noise was added to the x and y coordinates of every line endpoint of the translated and rotated image. Noise values were obtained by multiplying a random variable – uniformly distributed on the interval $[-1/2, 1/2]$ – by the noise magnitude parameters given in figures 2 and 3. For the second experiment input graphs were constructed by randomly selecting 20 lines from the reference images before rotating, translating and adding noise.

We compared the performance of the AUGM algorithm to the performance of the non-iterative probabilistic method for contextual correspondence matching of Kittler, Petrou and Christmas (KPC) [8] since it is one of the most well-known single-pass methods available, and similar to the AUGM algorithm, has a probabilistic origin. The AUGM is also compared to Bayesian Successive Projection Graph Matching (BAYSPGM) algorithm described in [17] since it is a single-pass technique derived by the authors which preceded and inspired AUGM methodology. The estimated probability of correct vertex-vertex assignments are reported in figures 2 and 3. These results were calculated for a given value of noise magnitude by averaging the results from 200 trials. All algorithms were implemented using the Gaussian compatibility function described in Ref. [6] (with a diagonal covariance matrix) and the Faugeras-Price (FP) compatibility function described in [16]. Although both compatibility functions were implemented for all three algorithms only the best results obtained are reported. For the results reported in figure 2 the AUGM algorithm was implemented using a Gaussian compatibility function, and the KPC and BAYPGM algorithms using the FP compatibility function. For the results reported in figure 3 the KPC and AUGM algorithms were implemented using a Gaussian compatibility function and the BAYSPGM algorithm using the FP compatibility function. From our results we conclude that (for the application considered in this paper and the given compatibility functions) the AUGM algorithm performed significantly better than the BAYSPGM algorithm, and that it performed slightly better than the KPC algorithm for full-graph matching case.

6 Conclusion

A joint probabilistic framework was presented that transforms the GM problem into a *Linear Assignment Problem* that was solved in an efficient manner using the Bertsekas auction algorithm. From the derivation of the joint probabilistic framework it is clear that the formulation inherently makes provision for input graphs with missing vertices. A strategy for handling spurious edges in the input graphs and incorporating vertex features has been devised but due to

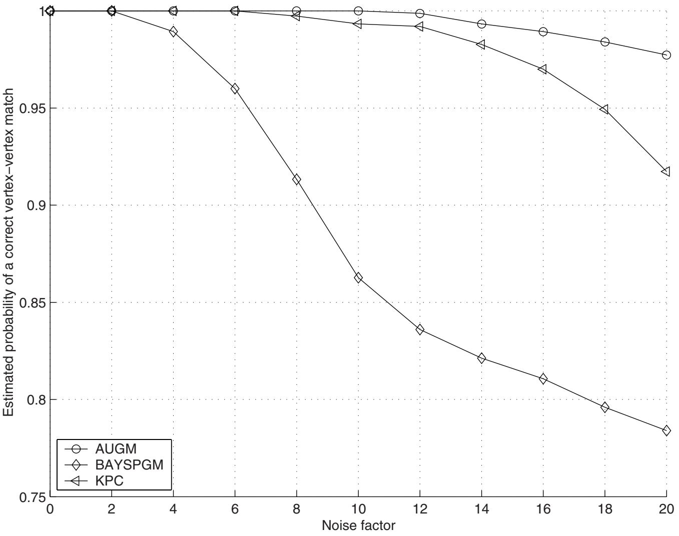

Fig. 2. Matching 50 translated and rotated input lines to 50 reference lines: Estimated probability of a correct vertex-vertex matching versus noise magnitude.

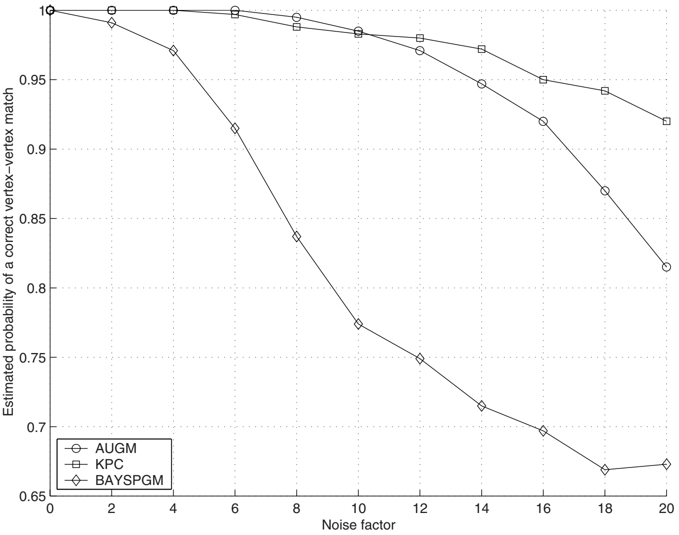

Fig. 3. Matching 20 translated and rotated input lines to 50 reference lines: Estimated probability of a correct vertex-vertex matching versus noise magnitude.

page constraints is beyond the scope of this paper. Work in progress include the derivation of a hybrid auction algorithm to directly solve the graph matching problem without first transforming it to a linear assignment problem.

References

1. Gold S. and Rangarajan A., A Graduated Assignment Algorithm for Graph Matching, IEEE Trans. Patt. Anal. Machine Intell, 18(4)(1996)377–388.
2. Simić P.D., Constrained Nets for Graph Matching and Other Quadratic Assignment Problems, Neural Computation, 3(1991)268–281.
3. Van Wyk B.J., Kronecker Product, Successive Projection, and Related Graph Matching Algorithms, PhD Thesis, University of the Witwatersrand, Johannesburg, 2003. [http://www.ee.wits.ac.za/comms/output/theses.htm]
4. Bertsekas D.P., The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial, Interfaces, 20(1990) 133–149.
5. Jonker R. and Volgenant A., A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems, Computing, 38(1987)325–340.
6. Christmas W.J., Kittler J. and Petrou M., Structural Matching in Computer Vision using Probabilistic Relaxation, IEEE Trans. Patt. Anal. Machine Intell., 17(8)(1995)749–764.
7. Cross A.D.J. and Hancock E.R., Graph Matching with a Dual Step EM Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11)(1998)1236–1253.
8. Kittler J., Petrou M. and Christmas W.J., A Non-iterative Probabilistic Method for Contextual Correspondence Matching, Pattern Recognition, 31(1998) 1455-1468.
9. Finch A.M., Wilson R.C. and Hancock R., Symbolic Matching with the EM Algorithm, Pattern Recognition, vol. 31(11)(1998)1777–1790.
10. Williams M.L., Wilson R.C. and Hancock E.R., Multiple Graph Matching with Bayesian Inference,Pattern Recognition Letters, 18(1997)1275–1281.
11. Cross A.D.J. and Hancock E.R., Graph Matching with a Dual Step EM Algorithm, IEEE Trans. Patt. Anal. Machine Intell., 20(11)(1998)1236–1253.
12. Wilson R.C. and Hancock E.R., A Bayesian Compatibility Model for Graph Matching, Pattern Recognition Letters, 17(1996)263–276.
13. Wilson R.C. and Hancock E.R., Structural Matching by Discrete Relaxation, IEEE Trans. Patt. Anal. Machine Intell., 19(6)(1997)634–648.
14. Finch A.M. , Wilson R.C. and Hancock E.R. , Matching Delauney Triangulations by Probabilistic Relaxation, Proceedings CAIP, (1995)351–358.
15. Meyers R.M., Wilson R.C. and Hancock E.R., Bayesian Graph Edit Distance,IEEE Trans. Patt. Anal. Machine Intell., 2(6)(2000)628–635.
16. Faugeras O.D. and Price K.E., Semantic Description of Aerial Images Using Stochastic Labeling, IEEE Transactions on Pattern Analysis and Machine Intelligence, 3(6)(1981)633-642.
17. Van Wyk B.J., Van Wyk M.A. and Botha J.J., A Matching Framework Based On Joint Probabilities, Proceedings of PRASA 2003, Langebaan, South Africa, 27-28 Nov. 2003, 125-130.
18. Caelli T. M. and Caetano T., Recent Developments in the Extraction and Matching of Image Structure and Syntax: From Relaxation to Junction Tree Models, Proceedings of PRASA 2003, Langebaan, South Africa, 27-28 Nov. 2003, 1-8.
19. Li S.Z., Matching: Invariant to Translations Rotations and Scale Changes, Pattern Recognition, 25(6)(1992)583-594.