A Matching Framework Based on Joint Probabilities

B.J. van Wyk, M.A. van Wyk, J.J. Botha

Kentron Dynamics and the Rand Afrikaans University

ben.van.wyk@kentron.co.za mavw@ing.rau.ac.za jaco.botha@kentron.co.za

Abstract

A *Single-Pass Joint Probabilistic* (SPJP) framework for contextual correspondence matching is presented. A salient feature of the SPJP approach is that inferred probabilities are not only constrained over all objects in the reference image, but are also constrained over all objects in the input image.

Keywords: Contextual Correspondence Matching, Joint Probabilistic Techniques, Graph Matching

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 problem. The focus of this contribution is the derivation of a *Single-Pass Joint Probabilistic* (SPJP) framework for graph matching.

A number of researchers proposed probabilistic frameworks for graph-based contextual correspondence matching: Peleg [1] was the first to derive an analytical updating rule for correspondence matching based on elementary probability theory. However, the work of Christmas *et al.* [2] on the theoretical underpinning of probabilistic relaxation, using the Bayesian framework, signalled a breakthrough. A non-iterative extension to the work of Christmas *et al.* was presented by Kittler *et al.* [4]. Finch *et al.* [10] on the other hand proposed a Bayesian framework for matching based on Delauney triangulations. Williams *et al.* [6] applied this methodology to multiple graph matching with Bayesian inference and an extension was presented by Finch *et al.* In Finch *et al.* [5] and Cross and Hancock [7] the graph matching process is described as a two step EM-like process, but once again the probability models underpinning their method were computed using a simple model of uniform matching errors and only sparse graphs were matched.

The framework presented in this paper differs significantly from previous probabilistic approaches for contextual correspondence and graph matching in the following aspects: *(1)* Instead of directly inferring $P(\theta_i = \overline{\theta}_{k,})$, the probabilities of associating a vertex in the input graph with a vertex in the reference graph, our main focus is on the inference 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.* [4], the SPJP approach is non-iterative. *(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 SPJP approach *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 SPJP framework does not rely on the use of compatibility functions specified in terms of the face-units of the graph under match or Bayesian edit distances based on these notions.

2. Notation

Say 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

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

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

Equations 1 to 3 represent the enforcement of two way assignment constraints in a probabilistic framework where the $P(\theta_i = \overline{\theta}_k)$ aren't restricted to assume binary values. 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

$$p(A_{ij} | \theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \tag{4}$$

corresponds to the compatibility coefficients calculated between edges of the input and references graphs using **edge** attributes. See for example Kittler *et al.* [4] , Christmas *et al.* [2] or Faugeras and Price [12]. We will now investigate how Eq. 4 relates to the joint probabilities

$$P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l). \tag{5}$$

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 $P(\theta_i =$

$\overline{\theta}_k, \theta_j = \overline{\theta}_l)$ using a novel optimization procedure 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 undefined 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

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

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.

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

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

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

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

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

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:

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

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 it can be shown that the sets $\left\{P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)\right\}_{k,l}$ (satisfying the constraints given by Eqs. 8 and 10) and $\left\{P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)\right\}_{i,j}$ (satisfying the constraints given by Eqs. 9 and 10) are both closed and convex, an easily implementable *Projection Onto Convex Set* (POCS) methodology can be used to solve Eq. 11. The following observation was used in our formulation to reduce complexity: Since $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_j = \overline{\theta}_l, \theta_i = \overline{\theta}_k)$ when our graphs are undirected, not all the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ need to enter Eq. 11. In fact we only need to consider the indices in the set

$$\{i,j,k,l\}_{j=i+1,\ldots,n \quad l=k+1,\ldots,\bar{n}} \qquad (12)$$

where $i = 1, \ldots, n$ and $k = 1, \ldots, \bar{n}$. Our optimization problem then becomes

$$\min_{P(\theta_i = \bar{\theta}_k, \theta_j = \bar{\theta}_l)} \sum_{i,j,k,l,\ i \neq j, k \neq l} \left(\frac{p(A_{ij}|\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{2} - P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) \right)^2 \qquad (13)$$

subject to Eqs. 8 to 10 where $i = 1, \ldots, n$, $k = 1, \ldots, \bar{n}$ and

$$i,j,k,l \in \{i,j,k,l\}_{j=i+1,\ldots,n \quad l=1,\ldots,\bar{n}}$$

in Eq. 13. The pseudo-code for a constraint enforcing POCS methodology, derived in [13] for a different purpose, which solves Eq. 13 is given by algorithms 1 and 2. $P_{i,j,k,l}$ denotes $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ in algorithm 1.

Algorithm 1 *Solving Eq. 13*:

begin initialize:
$k = 1,\ \delta > 10^{-3}$
$\mathbf{p} = [P_{1,2,1,2}, \ldots, P_{1,n,1,2}, P_{2,3,1,2}, \ldots, P_{n-1,n,1,2},$
$\quad P_{1,2,1,3}, \ldots, P_{1,n,1,3}, P_{2,3,1,3}, \ldots, P_{n-1,n,1,3}, \ldots,$
$\quad P_{1,2,\bar{n}-1,\bar{n}}, \ldots, P_{1,n,\bar{n}-1,\bar{n}}, P_{2,3,\bar{n}-1,\bar{n}}, \ldots, P_{n-1,n,\bar{n}-1,\bar{n}}]$
$\eta = \sum_{i=1}^{n-1}(n-i)$
$\bar{\eta} = \sum_{k=1}^{\bar{n}-1}(\bar{n}-k)$
while $\left(\delta > 10^{-3}\right.$ and $\left.k < \bar{\eta}\right)$
$\quad \mathbf{p}_h = \mathbf{p}$
$\quad \mathbf{p} \leftarrow T_r(\mathbf{p})$
$\quad \mathbf{p} \leftarrow T_c(\mathbf{p})$
$\quad \delta = (\mathbf{p}_h - \mathbf{p})^T (\mathbf{p}_h - \mathbf{p})$
$\quad k \leftarrow k + 1$
end while
end

The notation $\mathbf{p}(i : j : k)$ indicates that we select every j–th element from the vector $\mathbf{p}$, starting with the i–th element and ending with the k–th element. The operation $[\mathbf{s}, \mathbf{d}] = \text{sort}[\bar{\mathbf{p}}]$ indicates that we sort the elements of $\bar{\mathbf{p}}$ in ascending order, where $\mathbf{s}$ is the sorted vector and $\mathbf{d}$ is a vector containing the pre-sorted positions of the sorted elements.

Algorithm 2 *Pseudo-code for calculating* $T_r(\mathbf{p})$:

begin
for $i = 1 : \eta$
$\quad \phi = \text{sum}[\mathbf{p}(i : \eta : \eta(\bar{\eta} - 1) + i)]$
$\quad \sigma = \bar{\eta}$
$\quad \bar{\mathbf{p}} = \mathbf{p}(i : \eta : \eta(\bar{\eta} - 1) + i)$
$\quad [\mathbf{s}, \mathbf{d}] = \text{sort}[\bar{\mathbf{p}}]$
$\quad$ **for** $j = 1 : \bar{\eta}$
$\quad\quad \mathbf{s}(j) \leftarrow \mathbf{s}(j) + \frac{1-\phi}{\sigma}$
$\quad\quad$ **if** $\mathbf{s}(j) < 0$
$\quad\quad\quad \mathbf{s}(j) = 0$
$\quad\quad\quad \phi \leftarrow \phi - \bar{\mathbf{p}}(\mathbf{d}(j))$
$\quad\quad\quad \sigma \leftarrow \sigma - 1$
$\quad\quad$ **end if**
$\quad\quad \bar{\mathbf{p}}(\mathbf{d}(j)) = \mathbf{s}(j)$
$\quad$ **end for**
$\quad \mathbf{p}(i : \eta : \eta(\bar{\eta} - 1) + i) = \bar{\mathbf{p}}$
end for
end

When $\eta = \bar{\eta}$, the approach used to calculate $T_c(\mathbf{p})$ is similar to that which we used to calculate $T_r(\mathbf{p})$. The next algorithm is for the case $\eta < \bar{\eta}$:

Algorithm 3 *Calculating $T_c(\mathbf{p})$ when $\eta < \bar{\eta}$:*

begin
for $j = 1 : \bar{\eta}$
    $\phi = \text{sum}\left[\mathbf{p}(\eta(j-1)+1 : \eta j)\right]$
    if $\phi > 1$
        $\sigma = \eta$
        $\bar{\mathbf{p}} = \mathbf{p}(\eta(j-1)+1 : \eta j)$
        $[\mathbf{s}, \mathbf{d}] = \text{sort}\left[\bar{\mathbf{p}}\right]$
        for $i = 1 : \eta$
            $\mathbf{s}(i) \leftarrow \mathbf{s}(i) + \frac{1-\phi}{\sigma}$
            if $\mathbf{s}(i) < 0$
                $\mathbf{s}(i) = 0$
                $\phi \leftarrow \phi - \bar{\mathbf{p}}(\mathbf{d}(i))$
                $\sigma \leftarrow \sigma - 1$
            end if
            $\bar{\mathbf{p}}(\mathbf{d}(i)) = \mathbf{s}(i)$
        end for
        $\mathbf{p}(\eta(j-1)+1 : \eta j) = \bar{\mathbf{p}}$
    end if
end for
end

3.2. 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 *Given that the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ satisfy constraints 8 to 10 and assuming all $P(\theta_i = \overline{\theta}_k)$ satisfy constraints 1 to 3, then since the independence assumption holds (i.e. $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_k)P(\theta_j = \overline{\theta}_l)$) the probabilities $P(\theta_i = \overline{\theta}_k)$ are inferred by the summation process*

$$P(\theta_i = \overline{\theta}_k) = \frac{\sum_{j,l,j\neq i,l\neq k} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)}{n-1}. \tag{14}$$

Proof: Observe that by using the independence assumption it follows that

$$\sum_{j,l} P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l) = P(\theta_i = \overline{\theta}_k)\left(\sum_{j,l} P(\theta_j = \overline{\theta}_l)\right), \tag{15}$$

where $j \neq i$ and $l \neq k$. However, by assumption

$$\sum_{l} P(\theta_j = \overline{\theta}_l) = 1 \tag{16}$$

and Eq. 14 directly follows.

Proposition 2 *The probabilities $P(\theta_i = \overline{\theta}_k)$ obtained using proposition 1 will satisfy constraints 1 to 3 if the joint probabilities $P(\theta_i = \overline{\theta}_k, \theta_j = \overline{\theta}_l)$ satisfy constraints 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 $\bar{n} \geq n$.

3.3. 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

$$\max_{\overline{\theta}_k}\{P\left(\theta_i = \overline{\theta}_k\right)\}_{k=1,\ldots,\bar{n}}. \tag{17}$$

Except when there are many edges with identical attribute sets, it is highly unlikely that the same $\overline{\theta}_k$ will be assigned to more than one θ_i as $P(\theta_i = \overline{\theta}_k)$ will satisfy constraints 1 to 3.

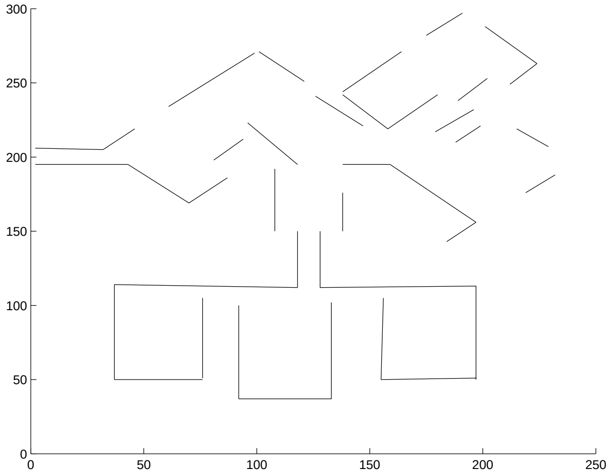

Figure 1: Thirty nine lines extracted from a simulated rooftop image.

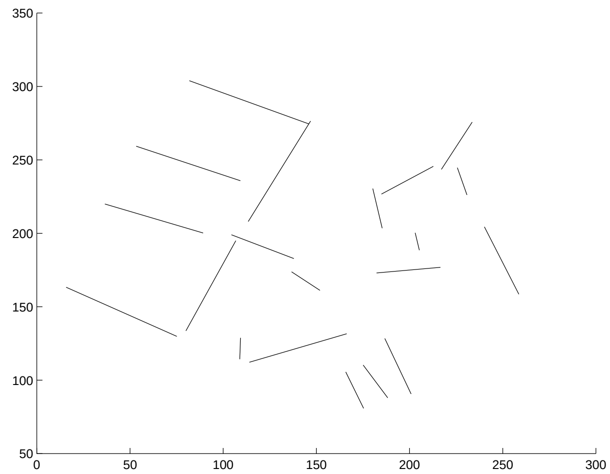

Figure 2: Twenty lines selected at random from the rotated and translated reference image with a noise factor of 10.

4. Simulation Results

To test the performance of the SPJP framework a line matching experiment was conducted. Rotated and translated versions of figure 1, such as figure 2, were matched to the original image. Lines were used to represent the vertices of a graph: 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]

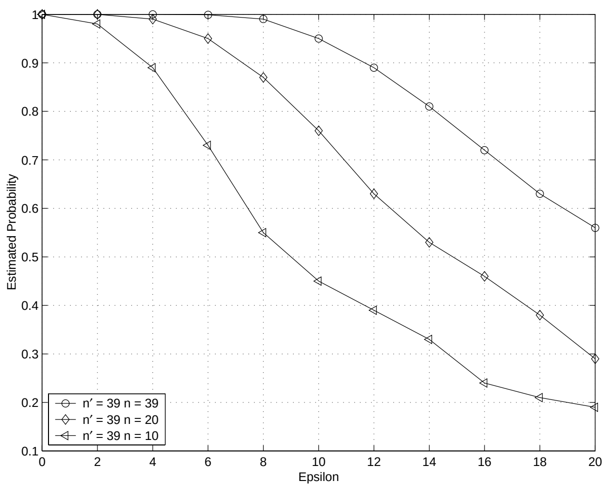

Figure 3: Matching 39, 20 and 10 translated and rotated input lines to 39 reference lines using the **single-pass** SPJP algorithm: Estimated probability of a correct vertex-vertex matching versus noise magnitude.

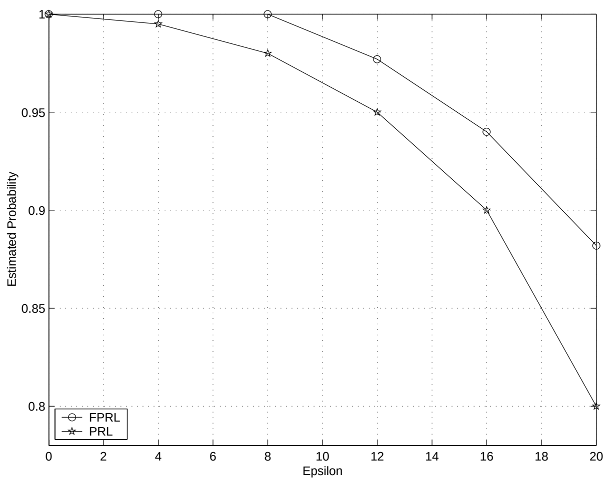

Figure 4: Matching 20 translated and rotated input lines to 39 reference lines using **multi-pass** techniques: Estimated probability of a correct vertex-vertex matching versus noise magnitude.

for more information on the derivation of translation and orientation invariant line features. For all our experiments best results were obtained using the compatibility function proposed by Faugeras and Price [12] with a roll-off parameter of 0.1 to estimate the conditional density functions.

For our experiment, all 39 lines depicted in figure 1 were used to construct a reference graph. Duplicate graphs having 39, 20 and 10 vertices were constructed using a random selection of rotated and translated lines extracted from a translated and rotated versions of figure 1. To test the robustness of the algorithms against line endpoint anomalies, 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 specified in table 1. Every line endpoint was therefore displaced by a value in the interval $[0, \rho]$ in an arbitrary direction, where ρ is the maximum line endpoint shift given by table 1. Refer to figure 2 for an example of an input image with a noise factor of 10.

The estimated probability of a correct vertex-vertex assignment reported in figure 3 was calculated for a given value of noise magnitude by averaging the results from 50 trials. As expected, algorithms incorporating a **multi-pass** approach and which in addition utilize vertex features outperform the SPJP algorithm. The performances of the Faugeras-Price Relaxation Labelling (FPRL) algorithm described in Ref. [12] and the *Probabilistic Relaxation Labelling* (PRL) algorithm of Christmas *et al.* [2] when matching 20 input lines to 39 reference lines are shown in figure 4. Line length in pixels was used as an additional vertex feature for each line and best results were obtained using the Faugeras and Price [12] compatibility function.

5. Conclusion

A novel probabilistic framework for performing attributed full- and sub-graph matching was presented. The SPJP approach is highly suited to contextual correspondence problems where an iterative algorithm is undesirable. From the derivation of the joint probabilistic framework it is clear that the formulation inherently makes provision for input graphs with missing vertices. Future work will concentrate on strategies for handling spurious edges in the input graphs. A possible drawback of the proposed framework is that it is based on edge attributes only, making no provision for the incorporation of vertex attributes. Future work will also include the incorporation of vertex attributes.

6. References

[1] S. Peleg , A New Probabilistic Relaxation Scheme, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2(4)(1980)362-369.

[2] W.J. Christmas, J. Kittler, M. Petrou, “Structural Matching in Computer Vision using Probabilistic Relaxation, IEEE Trans. Patt. Anal. Machine Intell., 17(8)(1995)749–764.

[3] 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(11)(1998)1236–1253.

[4] Kittler J., Petrou M., Christmas W.J.: A Non-iterative Probabilistic Method for Contextual Correspondence Matching, Pattern Recognition, 31(1998)1455-1468.

[5] Finch A.M., Wilson R.C., Hancock R. , Symbolic Matching with the EM Algorithm, Pattern Recognition, vol. 31(11)(1998)1777–1790.

[6] Williams M.L., Wilson R.C., Hancock E.R., Multiple Graph Matching with Bayesian Inference,Pattern Recognition Letters, 18(1997)1275–1281.

[7] Cross A.D.J., Hancock E.R. , Graph Matching with a Dual Step EM Algorithm, IEEE Trans. Patt. Anal. Machine Intell., 20(11)1998.

[8] Wilson R.C.,Hancock E.R., A Bayesian Compatibility Model for Graph Matching, Pattern Recognition Letters, 17(1996)263–276.

[9] Wilson R.C. , Hancock E.R. , Structural Matching by Discrete Relaxation, IEEE Trans. Patt. Anal. Machine Intell.,, 19(6)(1997)634–648.

[10] Finch A.M. , Wilson R.C. and Hancock E.R. , Matching Delauney Triangulations by Probabilistic Relaxation, Proceedings CAIP, (1995)351–358.

[11] 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.

[12] 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.

[13] 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]

[14] Van Wyk B.J. and Van Wyk M.A., The Spherical Approximation Graph Matching Algorithm, Proc. Int. Workshop on Multidisciplinary Design Optimisation, Pretoria, South Africa, 7-10 August, (2000)280–288.

[15] Van Wyk M.A., Durrani T.S., Van Wyk B.J., A RKHS Interpolator-Based Graph Matching Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7)(2002)988-995.

[16] Van Wyk B.J., Van Wyk M.A., Non-Bayesian Graph Matching without Explicit Compatibility Calculations, in Lecture Notes on Computer Science Vol. 2396: Structural, Syntactic, and Statistical Pattern Recognition, Caelli T., Amin A., Duin R.P.W., Kamel M. and de Ridder D. (ed.), Springer-Verlag, 2002(74–82)

[17] Van Wyk B.J., Van Wyk M.A., Kronecker Product Graph Matching, Pattern Recognition Journal, 36(2003)2019-2030.

[18] Van Wyk B.J., Van Wyk M.A. and Hanrahan H.E., Kronecker Match Matrix Estimation Algorithms for Graph Matching, Proceedings of the 12^{th} Annual Symposium of the Pattern Recognition Association of South Africa, Franschoek, South Africa, 29-30 November, (2001)53–58.

[19] Li S.Z., Matching: Invariant to Translations Rotations and Scale Changes, Pattern Recognition, 25(6)(1992)583-594.