Short Papers

A RKHS Interpolator-Based Graph Matching Algorithm

Michaël A. van Wyk, *Member*, *IEEE*,
Tariq S. Durrani, *Fellow*, *IEEE*, and
Barend J. van Wyk

Abstract—In this paper, we present a novel algorithm for performing attributed graph matching. This algorithm is derived from a generalized framework for describing functionally expanded interpolators [1] which is based on the theory of reproducing kernel Hilbert spaces. The algorithm incorporates a general approach to a wide class of graph matching problems based on attributed graphs, allowing the structure of the graphs to be based on multiple sets of attributes. No assumption is made about the adjacency structure of the graphs to be matched.

Index Terms—Graph matching, attributed relational graphs, reproducing kernel Hilbert space theory, combinatorial optimization, neural networks, pattern matching, image processing.

1 Introduction

One of the most general, as well as, natural approaches to the description of objects is in terms of their parts, the properties of these parts and their mutual relationships. This is referred to as the *structural description* of objects. Aspects associated with the structural description of objects are, for example, the construction of a structural description from acquired data, classification of structural descriptions, matching of descriptions, etc. One of the most challenging of these issues, is the optimal matching of structural descriptions. Matching in this context refers to finding a correspondence between the parts of different objects for the purpose of making the corresponding properties and relations as consistent as possible. This problem appears in image processing where it is required to match different images of the same object or similar objects based on the structural descriptions constructed from these images. Similarly matching techniques are useful for identification and classification of complex dynamical behaviors exhibited by nonlinear dynamical systems [2].

A structural description can be represented efficiently by means of an attributed (relational) graph. This enables us to interpret the problem of matching structural descriptions as the problem of matching two attributed graphs. The graph matching problem is of a combinatorial nature and can be solved by exhaustive search for the smallest of graphs only. Several algorithms have been proposed to solve graph matching problems in recent years. Graph matching algorithms can be divided into two major groups. In general, the first approach constructs a state-space which is searched using heuristic schemes to reduce complexity. Examples of algorithms belonging to this group are those proposed by You and Wong [4], Tsai and Fu [5], [6], Depiero et al. [7], Eshera and Fu [8], Bunke and Shearer [9] Bunke and Messmer [10], and Allen et al. [11].

- *M.A. van Wyk and B.J. van Wyk are with the Cybernetics Laboratory, Rand Afrikaans University, PO Box 524, Auckland Park, 2006 Gauteng, South Africa. E-mail: mavw@ing1.rau.ac.za.*
- *T.S. Durrani is with the Department of Electrical and Electronic Engineering, University of Strathclyde, 205 George St., Glasglow G1 1XW, Scotland, UK.*

Manuscript received 10 Nov. 1999; revised 9 Sept. 2000; accepted 14 Aug. 2001.
Recommended for acceptance by S.J. Dickinson.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110917.

The second approach is essentially based on nonlinear optimization techniques and includes algorithms such as the symmetric *Polynomial Transform Graph Matching* (PTGM) algorithm of Almohamad [12], the *Linear Programming Graph Matching* (LPGM) algorithm of Almohamad and Duffuaa [13], the *Eigen-decomposition Graph Matching* (EGM) method of Umeyama [14], the Lagrangian relaxation method of Rangarajan and Mjolsness [15], genetic algorithms [16], and a multitude of neural networks [17], [18], [19], [20], [21], [22], [23], [24] and relaxation based methods [26], [27], [28], [29], [30], [31],[32], [33]. Recently, Gold and Rangarajan [34] introduced the *Graduated Assignment Graph Matching* (GAGM) algorithm which proved to be a very successful algorithm. This algorithm combines graduated nonconvexity, two-way assignment constraints, and sparsity. The literature on optimization methods for graph matching includes the work of Hancock and his associates [35], [36], [37], [38]. Their work builds on a relational consistency gauged by an exponential probability distribution. Other recent contributions include the *Conjugate Gradient Graph Matching* (CGGM) algorithm [39] and the *Spherical Approximation Graph Matching* (SAGM) algorithm [41]. Both of these algorithms are based on the *Approximate Least-Squares Graph Matching* (ALSGM) algorithm proposed by van Wyk and Clark [40] and van Wyk and van Wyk [41].

The tree matching problem, for which various solutions have been proposed [42], [43], [44], [45], [46] is related to the graph matching problem. In [43], Pelillo et al. showed that the tree matching problem can be cast into an indefinite quadratic program. The tree matching problem is a special case of the more general graph matching problem but is not explicitly addressed in this paper.

In this paper, we present a novel algorithm for solving the attributed graph matching problem. This algorithm is derived by considering the corresponding pairs of attribute adjacency matrices and attribute vectors of two attributed graphs as corresponding arguments (input) and results (output) of a multi-input, multi-output, multivariable mapping. The graph matching problem then becomes a parametric interpolator identification problem. To derive this mapping, we introduce an appropriate *Reproducing Kernel Hilbert Space* (RKHS) [53] and derive an expression for its reproducing kernel. This kernel is utilized by an interpolator, which is linear in the adjustable weights, to identify the mapping relating corresponding pairs of attribute adjacency matrices. This forms an instance of a RKHS-based interpolator, the general framework of which is presented in [1] and, hence, the name *RKHS Interpolator-Based Graph Matching* (RIGM) algorithm. The salient features of the RIGM algorithm are: 1) It makes no assumption about the adjacency structure of the graphs and, therefore, is suitable for use with a larger class of graph matching problems than is generally the case with graph matching algorithms, 2) its mathematical structure enables the matching of huge graphs (with more than 500 vertices) to be handled on a Personal Computer (PC) with little effort in contrast with most other graph matching algorithms.

The outline of the presentation is as follows: In Section 2, we briefly review the concept of an attributed graph and formulate the attributed graph matching problem. Section 3 reviews the RKHS interpolator theory which is required for deriving the attributed graph matching algorithm presented here. The derivation of this graph matching algorithm is presented in Section 4. Some numerical and experimental results obtained during the evaluation of this algorithm are presented in Section 5 followed by the conclusion in Section 6.

2 Graph Matching Preliminaries

We represent an attributed graph G having r edge attributes and s vertex attributes by the symbol $G = (V, E, \{\mathbf{A}_k\}_{k=1}^{r}, \{\mathbf{B}_l\}_{l=1}^{s})$, where V is the set of vertices of the graph G, E is its set of edges, $\mathbf{A}_k \in \mathbb{R}^{n\times n}$ is the edge attribute adjacency matrix associated with the

kth edge and $\mathbf{B}_l \in \mathbb{R}^{n\times 1}$ is the lth vertex attribute vector. The number of vertices of G is $n := |V|$. Clearly, a weighted graph is a special case of an attributed graph, namely, with $r = 1$ and $s = 0$. For the purpose of our presentation, we restrict our attention to *complete undirected graphs* which implies that each edge adjacency matrix is symmetric and contains zeros along the diagonal.

Let

$$G' = (V', E', \{\mathbf{A}'_k\}_{k=1}^{r}, \{\mathbf{B}'_l\}_{l=1}^{s}),$$

and

$$G = (V, E, \{\mathbf{A}_k\}_{k=1}^{r}, \{\mathbf{B}_l\}_{l=1}^{s}),$$

respectively, denote our *reference* graph and *duplicate* graph. The number of vertices of G' is $n' := |V'|$ and the number of vertices of G is $n := |V|$, where $n' \geq n$ by assumption. *Attributed Graph Matching* (AGM) refers to the process of matching each vertex of G uniquely with a vertex of G' such that the correspondence between attribute values are as consistent as possible. *Full-graph matching* refers to matching two graphs having the same number of vertices (i.e., $n' = n$) while *subgraph matching* refers to matching two graphs having different number of vertices (i.e., $n' > n$). G is *matched* to some subgraph of G' if there exists an $n \times n'$ permutation submatrix $\mathbf{P}_o$ such that

$$\mathbf{A}_i = \mathbf{P}_o\, \mathbf{A}'_k\, \mathbf{P}_o^T\,, \qquad k = 1, \ldots, r$$

and

$$\mathbf{B}_l = \mathbf{P}_o\, \mathbf{B}'_l\,, \qquad l = 1, \ldots, s\,.$$

By modeling nonidealities in the construction process of the graphs by additive noise, one finds that in practical applications the above relationships may more realistically be expressed as

$$\mathbf{A}_i = \mathbf{P}_o\, \mathbf{A}'_k\, \mathbf{P}_o^T + \epsilon \mathbf{N}_k\,, \qquad k = 1, \ldots, r\,, \tag{1}$$

and

$$\mathbf{B}_l = \mathbf{P}_o\, \mathbf{B}'_l + \epsilon \mathbf{M}_l\,, \qquad l = 1, \ldots, s\,, \tag{2}$$

where $\mathbf{N}_k$ is an $n \times n$ noise matrix, $\mathbf{M}_l$ is an $n \times 1$ noise vector, and ϵ is related to the noise power and is assumed to be independent of the indices k and l.

In the nonideal case, the AGM problem may then be expressed as the following optimization problem:

$$\min_{\mathbf{P}} \left(\sum_{k=1}^{r} W_k \left\| \mathbf{A}_k - \mathbf{P}\mathbf{A}'_k\mathbf{P}^T \right\|^q + \sum_{l=1}^{s} W_{l+r} \left\| \mathbf{B}_l - \mathbf{P}\mathbf{B}'_l \right\|^q \right), \quad \mathbf{P} \in \mathrm{Per}(n, n')\,, \tag{3}$$

where $\|\cdot\|$ represents some norm, $\mathrm{Per}(n, n')$ denotes the set of all $n \times n'$ permutation submatrices, and $\{W_k\}_{k=1}^{r+s}$ is a set of weights satisfying

$$0 \leq W_k \leq 1\,, \quad k = 1, \ldots, r+s \qquad \text{and} \qquad \sum_{k=1}^{r+s} W_k = 1\,.$$

Different norms will yield different solutions to the above minimization problem. Typical values for q are 1 and 2 . Note that no explicit distinction is made between the norm used for terms containing attribute adjacency matrices and the norm used for terms containing attribute vectors in (3). Also note that the ideal case ($\epsilon = 0$) is contained as a special case of this optimization problem. For the ideal case this is referred to as *exact matching*, whereas, for the nonideal case ($\epsilon > 0$), it is referred to as *inexact matching*.

3 RKHS Interpolator Theory Preliminaries [1]

3.1 Theoretical Framework

Suppose we wish to approximate a real-valued function unknown to us. We assume that its domain $\mathbb{X} \subseteq \mathbb{R}^N$ is a Borel set. Furthermore, we introduce the Borel set $\mathbb{U} \subset \mathbb{R}^N$ and require it to be a symmetric subset of some closed N-cube

$$[-\mathbf{U}, \mathbf{U}] := [-U_1, U_1] \times \cdots \times [-U_N, U_N]\,.$$

The purpose of the set $\mathbb{U}$ will become clear shortly.

We proceed by introducing two measure spaces [49] $(\mathbb{U}, \mathcal{B}_{\mathbb{U}}, \mu)$, and $(\mathbb{X}, \mathcal{B}_{\mathbb{X}}, \lambda)$, and let $\mathcal{B}^N$ be the σ-algebra generated by subsets of $\mathbb{R}^N$ of the form $A_1 \times \cdots \times A_N$, with each A_i a Borel subset of $\mathbb{R}$. Here, $\mathcal{B}_{\mathbb{U}}$ and $\mathcal{B}_{\mathbb{X}}$ denote the restrictions of $\mathcal{B}^N$ to Borel subsets $\mathbb{U}$ and $\mathbb{X}$, respectively. We restrict our attention to the case where the measure μ is the product measure induced by a σ-finite measure $\mu_1 : \mathcal{B}^1 \to [0, \infty]$.

Consider the vector space $\mathcal{V}_{\mathbb{U}}$ of all functions $\mathbf{F} : \mathbb{X} \to \mathbb{R}$ of the form

$$\mathbf{F}(\mathbf{x}) = \int_{\mathbb{U}} \varphi(\mathbf{u})\; e(\mathbf{u}, \mathbf{x})\; \mu(d\mathbf{u})\,, \tag{4}$$

where $\varphi \in L^2(\mathbb{U}, \mathcal{B}_{\mathbb{U}}, \mu)$ is called the *spectrum* of $\mathbf{F}$ and $e : \mathbb{U} \times \mathbb{X} \to \mathbb{C}$ is an element of $L^2(\mathbb{U} \times \mathbb{X}, \sigma(\mathcal{B}_{\mathbb{U}} \times \mathcal{B}_{\mathbb{X}}), \mu \times \lambda)$ such that the (*fundamental*) family $\mathcal{E} := \{e(\mathbf{u}, \cdot)\}_{\mathbf{u} \in \mathbb{U}}$ contains a *total* set of functions in $L^2(\mathbb{X}, \mathcal{B}_{\mathbb{X}}, \lambda)$. Additionally, we require that $e(-\mathbf{u}, \cdot) = e^*(\mathbf{u}, \cdot)$ holds μ-a.e. when e is complex-valued and similarly for φ. Using the Cauchy-Schwarz inequality and Fubini's theorem it follows readily that

$$|F(\mathbf{x})| \leq \|\varphi\|_{L^2}\, \|e(\cdot, \mathbf{x})\|_{L^2} < \infty$$

and, hence, every *evaluation functional* defined on $\mathcal{V}_{\mathbb{U}}$ is continuous.

Next, we introduce a measure $\nu << \mu$ with Radon-Nikodym derivative $v \geq 0$ with respect to μ and define $\mathcal{H}_{\mathbb{U}}$ to be the completion of the subspace of $\mathcal{V}_{\mathbb{U}}$ for which the inner product

$$(F, G)_{\mathcal{H}_{\mathbb{U}}} := \int_{\mathbb{U}} \varphi(\mathbf{u}) \gamma^*(\mathbf{u}) v(\mathbf{u})\; \mu(d\mathbf{u}) \tag{5}$$

is finite for each pair $F, G \in \mathcal{H}_{\mathbb{U}}$ with φ and γ their respective spectra (see (4)). Here, γ^* denotes the complex conjugate of γ. The measure ν is selected such that the space $\mathcal{H}_{\mathbb{U}}$ incorporates the available a priori knowledge of the specific approximation problem being considered. The space $(\mathcal{H}_{\mathbb{U}}, (\cdot, \cdot)_{\mathcal{H}_{\mathbb{U}}})$ is a RKHS with reproducing kernel $K : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ defined by

$$K(\mathbf{x}, \mathbf{y}) := \int_{v>0} \kappa_{\mathbf{x}}(\mathbf{u})\; e(\mathbf{u}, \mathbf{y})\; \mu(d\mathbf{u})\,, \qquad \kappa_{\mathbf{x}}(\cdot) := \frac{e^*(\cdot, \mathbf{x})}{v(\cdot)}\,, \tag{6}$$

subject to the conditions [1]

$$\int_{v>0} \left| \frac{e(\mathbf{u}, \mathbf{x})}{v(\mathbf{u})} \right|^2 \mu(d\mathbf{u}) < \infty \quad \text{and} \quad \int_{v>0} \frac{|e(\mathbf{u}, \mathbf{x})|^2}{v(\mathbf{u})}\; \mu(d\mathbf{u}) < \infty \tag{7}$$

for all $\mathbf{x} \in \mathbb{X}$. It is easily verified that $K(\cdot, \cdot)$ is symmetric in its arguments, positive definite, and satisfies the *reproducing property*, namely, $(F, K(\mathbf{x}, \cdot))_{\mathcal{H}_{\mathbb{U}}} = F(\mathbf{x})$. For convenience, we define the *weighting function* w as the reciprocal of the function v.

3.2 Parameter Identification Procedure

For the training set $\mathcal{T} := \{(\mathbf{x}_i, y_i)\}_{i=1}^{M_t}$ consisting of input-output pairs given, we briefly summarize the process of calculating the unknown parameters of the approximator. By assumption, these input-output pairs are related according to

$$(F, K(\mathbf{x}_i, \cdot))_{\mathcal{H}_{\mathbb{U}}} = F(\mathbf{x}_i) = y_i, \qquad i = 1, \ldots, M_t\,. \tag{8}$$

From Approximation Theory [53, p. 65], we recall that, the *minimum norm* $\mathcal{H}_{\mathbb{U}}$-approximation to the unknown nonlinear mapping $F|_{\mathbb{X}}$, say $\widetilde{F}$, satisfying the given set of *interpolative constraints*, i.e., $\widetilde{F}(\mathbf{x}_i) = y_i$, $(i = 1, \ldots, M_t)$ is of the form

$$\widetilde{F}(\cdot) = \sum_{l=1}^{M_t} C_l \, K(\mathbf{x}_l, \cdot), \tag{9}$$

where the real-valued coefficients $\{C_l\}_{l=1}^{M_t}$ are derived from the system of simultaneous equations that results from evaluating (9) at each of the *centers* $\{\mathbf{x}_k\}_{k=1}^{M_t}$. Observe that the approximant $\widetilde{F}$ is linear in the unknown parameters $\{C_l\}_{l=1}^{M_t}$. Requiring that the above expression for $\widetilde{F}$ satisfy the interpolative constraints produces a system of linear equations which is then solved analytically for the weights $\{C_l\}_{l=1}^{M_t}$.

3.3 Multivariate Polynomial Approximators

The particular instance of the above framework of interest here is the space of all real-valued multivariate polynomials of degree n in the real variables $\{x_i\}_{i=1}^{N}$. We use the symbol $\mathcal{V}_n$ to denote this space. An element F of this space is represented by (see [51])

$$F(\mathbf{x}) = \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{n}} f_{\mathbf{k}} \frac{\mathbf{x}^{\mathbf{k}}}{\mathbf{k!}} \, ,$$

where $(f_{\mathbf{k}}) \in \mathbb{R}^{(n+1)^N}$, $\mathbf{k} := (k_1, k_2, \ldots, k_N)$, $\mathbf{n} := n1$, $\mathbf{x} := (x_i)$, and

$$\mathbf{1} := (1, \ldots, 1)' \, , \qquad \mathbf{k!} := k_1! \, k_2! \, \cdots \, k_N! \, ,$$
$$|\mathbf{k}| := k_1 + k_2 + \cdots \, k_N \, , \quad \mathbf{x}^{\mathbf{k}} := x_1^{k_1} \, x_2^{k_2} \, \cdots \, x_N^{k_N} \, .$$

With the members of $\mathcal{V}_n$, the RKHS space $\mathcal{H}_n$ is constructed by introducing the inner product [1], [51],

$$(F, G)_{\mathcal{H}_n} := \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{n}} \frac{\rho_{\mathbf{k}}}{\mathbf{k!}} f_{\mathbf{k}} g_{\mathbf{k}} \, ,$$

where, as explained in [51], the sequence $(\rho_{\mathbf{k}})$ is selected based on a priori knowledge of the problem under consideration. The corresponding reproducing kernel is given by

$$K(\mathbf{x}, \mathbf{y}) := \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{n}} \frac{1}{\mathbf{k!} \, \rho_{\mathbf{k}}} \, \mathbf{x}^{\mathbf{k}} \mathbf{y}^{\mathbf{k}} \, .$$

4 Interpolator-Based Graph Matching Algorithm

Suppose we are required to match two attributed graphs G' and G. As before, we consider G' to be the reference graph and G the duplicate graph. To simplify the presentation later on in this section, we now introduce some notation. The attribute adjacency matrices and attribute vectors of both graphs are expressed as

$$\mathbf{A}_k := (A_{k\,|\,i\,j}), \qquad \mathbf{A}'_k := (A'_{k\,|\,i\,j})$$

and

$$\mathbf{B}_l := (B_{l\,|\,i}), \qquad \mathbf{B}'_l := (B'_{l\,|\,i}) \, ,$$

where $k = 1, \ldots, r$ and $l = 1, \ldots, s$.

In order to simplify the presentation of the RIGM algorithm, we first make the observation that vertex attribute vectors converted to diagonal matrices, using the $\mathrm{diag}(\cdot)$ operation in linear algebra, satisfy the same expression as edge attribute matrices do, namely,

$$\mathrm{diag} \, \mathbf{B}_l \approx \mathbf{P} \, \mathrm{diag}(\mathbf{B}'_l) \, \mathbf{P}^T \, ,$$

with exact equality holding for the ideal (i.e., noiseless) case. This means that these converted vertex attribute vectors may be considered as additional edge attribute matrices (although no longer of a complete graph) and, consequently, the AGM problem can now be expressed as

$$\min_{\mathbf{P}} \left(\sum_{k=1}^{r+s} W_k \| \mathbf{A}_k - \mathbf{P} \mathbf{A}'_k \mathbf{P}^T \|^q \right), \quad \mathbf{P} \in \mathrm{Per}(n, n').$$

We now view the AGM process comprising the mapping

$$\mathbf{F} : \mathbb{R}^{n' \times n'} \to \mathbb{R}^{n \times n} \, , \qquad \mathbf{F}(\mathbf{X}) := \mathbf{P} \, \mathbf{X} \, \mathbf{P}^T \, ,$$

with the fixed parameter matrix $\mathbf{P} \in \mathrm{Per}(n, n')$, the unknown quantity to be determined. In doing so, the AGM problem is cast into a system identification problem which can be solved by selecting an appropriate RKHS-based interpolator to model the above mapping. This interpolator contains implicitly an approximation to $\mathbf{P}$. The input-output pairs used for establishing the interpolative constraints associated with $\mathbf{F}$ is the set $\{(\mathbf{X}_k, \mathbf{Y}_k)\}_{k=1}^{r+s}$, where

$$\mathbf{X}_k := \mathbf{A}'_k \, , \qquad \mathbf{Y}_k := \mathbf{A}_k \, .$$

In component form, the mapping $\mathbf{F} = (F_{i\,j})$ can be expressed as

$$F_{i\,j}(\mathbf{X}) = \mathbf{p}_i^T \, \mathbf{X} \, \mathbf{p}_j \, ,$$

where $\mathbf{p}_i^T$ denotes the ith row vector of the matrix $\mathbf{P}$.

4.1 The Interpolator Expression

The material presented in Section 3 will now be used to find a solution to the AGM problem. Consider the space of all functions of the form

$$F : \mathbb{R}^{n'^2 \times 1} \to \mathbb{R} \, , \qquad F(\mathbf{X}) = \sum_{i,j=1}^{n'} \varphi_{i\,j} \, X_{i\,j} \, .$$

The component functions $F_{i\,j}$ of the function $\mathbf{F}$ clearly belong to this space. We choose as basis for this space the functions

$$e_{i\,j}(\mathbf{X}) := X_{i\,j} \, , \qquad i, j = 1, \ldots, n' .$$

In terms of the results of Section 3, this implies that

$$\rho_{\mathbf{k}} = \begin{cases} 1, & |\mathbf{k}| = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Next, we endow this space with the inner product

$$(F, G) := \sum_{i,j=1}^{n'} \varphi_{i\,j} \, \gamma_{i\,j} \, .$$

We now have a RKHS with reproducing kernel given by

$$K(\mathbf{Y}, \mathbf{Z}) = \sum_{i,j=1}^{n'} e_{i\,j}(\mathbf{Y}) \, e_{i\,j}(\mathbf{Z}) = (\mathrm{vec} \, \mathbf{Y})^T \, \mathrm{vec} \, \mathbf{Z} \, .$$

Here, $\mathrm{vec}(\cdot)$ is the vectorization operation of linear algebra. Given the training set $\{(\mathbf{A}'_k, \mathbf{A}_k)\}_{k=1}^{r+s}$ we therefore have the following interpolative constraints:

$$(F_{i\,j}, K(\mathbf{A}'_k, \cdot)) = F_{i\,j}(\mathbf{A}'_k) = A_{k\,|\,i\,j}$$

for each $k = 1, \ldots, r + s$ and $i, j = 1, \ldots, n$. We recall from Section 3 that the minimum-norm interpolator has the form

$$\widetilde{F}_{i\,j}(\cdot) = \sum_{l=1}^{r+s} C_{l\,|\,i\,j} \, K(\mathbf{A}'_l, \cdot) \, ,$$

where the coefficients $C_{l\,|\,i\,j}$ are the unknown parameters. Evaluation of the function $\widetilde{F}_{i\,j}$ at the points $\mathbf{X} = \mathbf{A}'_k$ yields the following system of simultaneous equations,

$$A_{k\,|\,i\,j} = \sum_{l=1}^{r+s} C_{l\,|\,i\,j}\, K(\mathbf{A}'_l, \mathbf{A}'_k)\ , \qquad k = 1, \ldots, r+s\ ,$$

for $i, j = 1, \ldots, n$. Assembling these results into a matrix, we obtain

$$\mathbf{A}_k = \sum_{l=1}^{r+s} \mathbf{C}_l\, G_{l\,k}\ , \qquad k = 1, \ldots, r+s\ ,$$

where $\mathbf{C}_l := (C_{l\,|\,i\,j})$ and $G_{l\,k} := K(\mathbf{A}'_l, \mathbf{A}'_k)$. This expression may also be written in the form

$$\begin{aligned}\operatorname{vec} \mathbf{A}_k &= \operatorname{vec}\left(\sum_{l=1}^{r+s} \mathbf{C}_l\, G_{l\,k}\right)\\ &= \sum_{l=1}^{r+s} \operatorname{vec}(\mathbf{C}_l) G_{l\,k}\\ &= \mathbf{C}\, \mathbf{G}_k\ ,\end{aligned}$$

with $\mathbf{C} \in \mathbb{R}^{n^2 \times (r+s)}$ and $\mathbf{G}_k \in \mathbb{R}^{(r+s)\times 1}$ defined by

$$\mathbf{C} := (\operatorname{vec} \mathbf{C}_1, \ldots, \operatorname{vec} \mathbf{C}_{r+s}) \ \text{ and } \ \mathbf{G}_k := \left(G_{1\,k}, \ldots, G_{r+s,\,k}\right)^T\ ,$$

respectively. By introducing the matrices $\mathbf{A} \in \mathbb{R}^{n^2 \times (r+s)}$ and $\mathbf{G} \in \mathbb{R}^{(r+s)\times(r+s)}$ defined as

$$\mathbf{A} := (\operatorname{vec} \mathbf{A}_1, \ldots, \operatorname{vec} \mathbf{A}_{r+s}) \ \text{ and } \ \mathbf{G} := (\mathbf{G}_1, \ldots, \mathbf{G}_{r+s})\ ,$$

we can express the complete problem in the form of a single matrix equation, namely,

$$\mathbf{A} = \mathbf{C}\, \mathbf{G}\ ,$$

where the only unknown is the matrix $\mathbf{C}$. Conditions under which the Gram matrix $\mathbf{G}$ is invertible are stated in [53, p. 56]. These conditions are assumed to be satisfied by the edge attribute matrices of an attributed graph and, hence, the coefficients of the interpolator are described by the matrix expression,

$$\mathbf{C} = \mathbf{A}\, \mathbf{G}^{-1}\ . \tag{10}$$

In summary, the expression that describes the approximating multi-input, multi-output, multivariable interpolator is

$$\widetilde{\mathbf{F}}(\cdot) = \sum_{l=1}^{r+s} \mathbf{C}_l\, K(\mathbf{A}'_l, \cdot)\ . \tag{11}$$

Remark. To simplify the above presentation, we included all $r+s$ terms in the interpolator. If, more generally, we wish to use $1 \le \widehat{r} \le r+s$ number of terms in the interpolator, we may do so, keeping in mind that we then also need to replace $\{\mathbf{A}'_l\}_{l=1}^{r+s}$ in the first argument of the reproducing kernel in the expression for the interpolator, (11), by kernel centers $\{\widetilde{\mathbf{A}}'_l\}_{l=1}^{\widehat{r}}$ derived by some means of some data reduction method [1]. Using the least-squares criterion as a measure of optimality, the inverse of $\mathbf{G}$ in (10) must then be replaced by the pseudoinverse of $\mathbf{G}$.

4.2 Parameter Matrix Inference via Optimal Assignment

Although, we have now derived an approximation to the unknown function, $\mathbf{F}$, we still need to extract the approximating permutation submatrix $\widetilde{\mathbf{P}}$ from it. This poses a problem since, as mentioned earlier, the approximant $\widetilde{\mathbf{F}}$ contains $\widetilde{\mathbf{P}}$ only implicitly and, therefore, we have to devise some scheme to infer $\widetilde{\mathbf{P}}$ from $\widetilde{\mathbf{F}}$. Our approach in this regard is based on the observation that for any $\mathbf{P} \in \operatorname{Per}(n, n')$ the following identities hold, namely,

$$\mathbf{P}\ \mathbf{1}'_{\sigma(i),:}\ \mathbf{P}^T = \mathbf{1}_{i,:} \qquad \text{and} \qquad \mathbf{P}\ \mathbf{1}'_{:,\sigma(j)}\ \mathbf{P}^T = \mathbf{1}_{:,j}\ , \tag{12}$$

where $i, j = 1, \ldots, n$ and σ is the permutation induced by the permutation submatrix $\mathbf{P}$. Here, $\mathbf{1}_{i,:}$ is an $n \times n$ matrix with each element of the ith row equal to 1 and all other elements equal to zero and $\mathbf{1}_{:,j} := \mathbf{1}_{j,:}^T$. Similarly, $\mathbf{1}'_{\sigma(i),:}$ is an $n' \times n'$ matrix with each element of the $\sigma(i)$th row equal to 1 and all other elements equal to zero and $\mathbf{1}'_{:,\sigma(j)} := \mathbf{1}'^T_{\sigma(j),:}$. It is obvious that the identities (12) can be used to recover the permutation submatrix $\mathbf{P}$. If $\mathbf{P}$ is the permutation submatrix that parameterizes the unknown mapping, $\mathbf{F}$, and σ is the permutation induced by $\mathbf{P}$, then the above identities are equivalent to

$$\mathbf{F}(\mathbf{1}'_{\sigma(i),:}) = \mathbf{1}_{i,:} \qquad \text{and} \qquad \mathbf{F}(\mathbf{1}'_{:,\sigma(j)}) = \mathbf{1}_{:,j}\ ,$$

where $i, j = 1, \ldots, n$. By replacing $\mathbf{F}$ by the approximant $\widetilde{\mathbf{F}}$, we expect these identities still to hold approximately, that is,

$$\widetilde{\mathbf{F}}(\mathbf{1}'_{\sigma(i),:}) \approx \mathbf{1}_{i,:} \qquad \text{and} \qquad \widetilde{\mathbf{F}}(\mathbf{1}'_{:,\sigma(j)}) \approx \mathbf{1}_{:,j}\ ,$$

for $i, j = 1, \ldots, n$. It is possible to use *both* these identities together to derive an approximation to $\mathbf{P}$. However, we only utilize the first one. By averaging the elements of the matrix $\widetilde{\mathbf{F}}(\mathbf{1}'_{i',:})$ row-wise for a given value of i' , we obtain a column vector which we then view as the i'th column of a matrix, say $\mathbf{W}$. After repeating this process for each $i' = 1, \ldots, n'$ and ensuring that all elements of $\mathbf{W}$ are nonnegative, adding a constant to $\mathbf{W}$ if necessary, we then use this matrix as the weight matrix of an *optimal assignment problem*, the solution of which is a permutation submatrix $\widetilde{\mathbf{P}}$ representing the optimal assignment. The matrix $\widetilde{\mathbf{P}}$ is an approximation to $\mathbf{P}$, the permutation submatrix which parameterizes the unknown mapping $\mathbf{F}$. It can be shown that $\widetilde{\mathbf{P}}$ is the permutation submatrix closest to $\mathbf{W}$ in the sense that $\|\widetilde{\mathbf{P}} - \mathbf{W}\|_F$ is a minimum. Here, $\|\cdot\|_F$ denotes the Frobenius norm. It is obvious for the ideal case that $\widetilde{\mathbf{P}} \equiv \mathbf{P}$.

5 Numerical Experiments

5.1 Numerical Experiments

In order to evaluate the performance of the RIGM algorithm, the following procedure was used: First, the parameters n', n , r, and s were fixed. For every iteration a reference graph G' was generated randomly with all attributes distributed uniformly between 0 and 1. An $n \times n'$ permutation submatrix, $\mathbf{P}$, was also generated randomly and, then, used to permute the rows and columns of the edge attribute adjacency matrices and the elements of the vertex attribute vectors of G'. Next, an independently generated noise matrix (vector, respectively) was added to each edge attribute adjacency matrix (vertex attribute vector, respectively) to obtain the duplicate graph G. The element of each noise matrix or vector was obtained by multiplying a random variable—uniformly distributed on the interval $[-1/2, 1/2]$—by the noise magnitude parameter ϵ. As described below, different graph matching algorithms were then used to calculate a permutation submatrix which approximates the original permutation submatrix $\mathbf{P}$. For each algorithm, by directly comparing the derived and original permutation submatrices, entry by entry, the probability of a correct vertex-vertex assignment was calculated for a given value of ϵ based on 500 trials. From a probabilistic point of view this reflects how well an algorithm performs for a given noise magnitude.

5.2 Numerical Results

The performance of the RIGM algorithm was compared to the performance of the GAGM, EGM, SAGM, CGGM, PTGM, and LPGM algorithms. The reasons for selecting these specific algorithms are because they do not make any assumption about the adjacency structures of the graphs and can handle attributed graphs with multiple attributes.

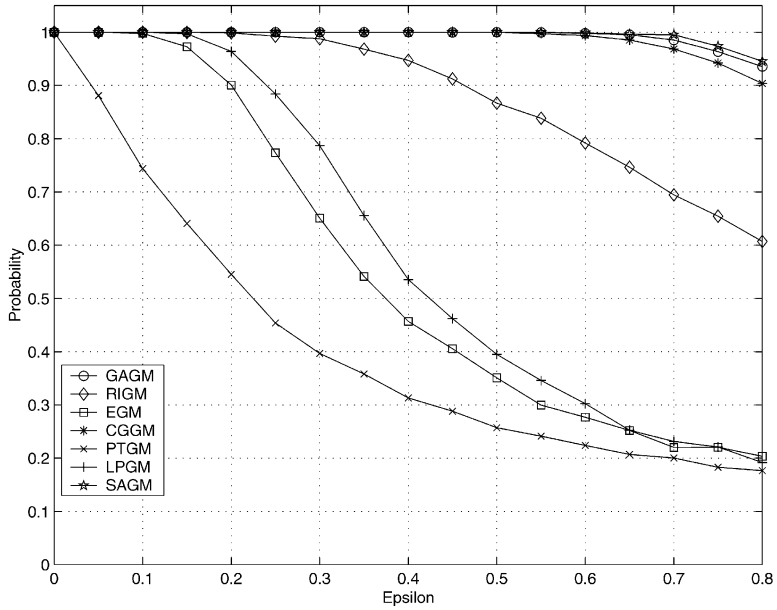

Fig. 1. Matching of $(10, 3, 3)$ attributed graphs: Probability of correct vertex-vertex matching versus ϵ.

Full-Graph Matching. Fig. 1 presents the probability of correct vertex-vertex matching as a function of noise magnitude ϵ for the case $(n, r, s) = (10, 3, 3)$. Globally, the performance curves for the SAGM, GAGM, and CGGM algorithms are closely spaced, and well separated from the performance curves of the LPGM, PTGM, and EGM algorithms for large values of ϵ. Although, the RIGM algorithm outperformed the LPGM, EGM, and PTGM algorithms for $\epsilon > 0.15$, it was outperformed by the SAGM, GAGM, and CGGM algorithms for $\epsilon > 0.3$.

Fig. 2 shows the results obtained for $(n, r, s) = (30, 3, 3)$. For values of ϵ up to 0.2, the performance of the RIGM algorithm was near ideal. In general, there was a decline in performance of *all* algorithms for this case. This was due to the fact that the number of attributes was kept constant while the number of vertices increased. In order to observe this phenomenon more clearly, we kept both r and s fixed to 3 and incremented the number of nodes, n , from 50 to 300 in steps of 50 (see Fig. 3). The results of this experiment is shown in Fig. 4. It is interesting to note that these curves converge in a monotonic decreasing manner to a *nonzero* asymptote as n increases. This is supported by the fact that the maximum separation between the curves corresponding to $n = 250$ and $n = 300$ is only 0.02 . Unfortunately, as explained in the discussion below, we could not verify this phenomenon for the other algorithms due to their storage requirements.

Subgraph Matching. For the purpose of assessing the RIGM algorithm's ability to perform subgraph matching, its performance was compared to that of the GAGM, CGGM, and SAGM algorithms only as the EGM, PTGM, and LPGM algorithms are not suited to subgraph matching. Fig. 4 depicts the probability of correct vertex-vertex matching for the case $(n/n', r, s) = (5/15, 5, 5)$. The results for this experiment indicate that the probability of a correct vertex-vertex match was greater than 0.8 for values of ϵ up to 0.3 when two

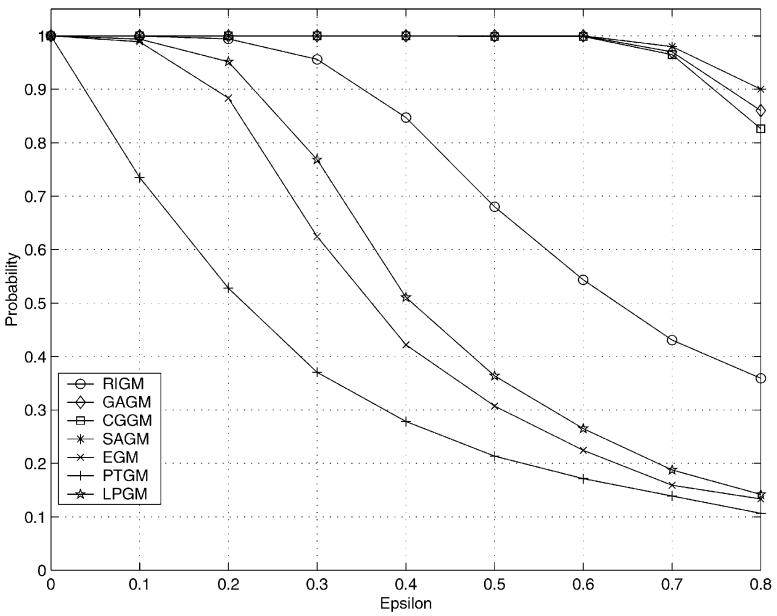

Fig. 2. Matching of $(30, 3, 3)$ attributed graphs: Probability of correct vertex-vertex matching versus ϵ.

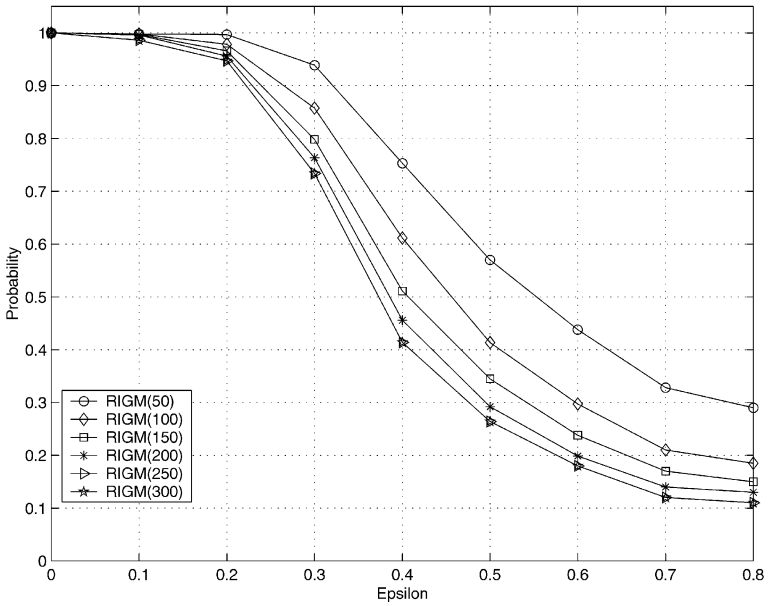

Fig. 3. Matching of $(n, 3, 3)$ attributed graphs for $n = 50,\ 100,\ \ldots\ , 300$: Probability of correct vertex-vertex matching versus ϵ.

thirds of the vertices were dropped from the duplicate graph. When only two or three vertices were missing, the experiment was trivial since the RIGM algorithm almost always found the correct vertex-vertex match in the noiseless case.

Discussion. Although, Figs. 1 and 2 indicate that, for large noise, the RIGM algorithm is outperformed by the CGGM, SAGM, and GAGM algorithms, its strength is its relatively small storage requirement. The GAGM, CGGM, and SAGM algorithms requires a matrix with $(nn')^2$ elements to be stored in memory whereas the largest matrix to be stored by the RIGM algorithm only has $rs(n')^2$ elements which can be reduced further for the case when $r = s$ by storing significant elements only. For example, for the case $(n, r, s) = (300, 3, 3)$, the GAGM, CGGM, and SAGM algorithms require the storage of about 8.1×10^9 elements, whereas the RIGM algorithm only requires the storage of 5.4×10^5 elements. The large storage requirements of algorithms such as the GAGM algorithm prevented us from performing proper comparison and characterization experiments for graphs with more than 50 nodes using a Pentium III 800 PC with 128 Mb of RAM running MATLAB. The reduced storage requirement of the RIGM algorithm is a major advantage when implementing a real-time system where the memory requirement is a critical factor. Single runs for graphs with up to a 500 vertices were conducted on a PC without experiencing any problems.

In addition to the reduced storage requirement of the RIGM algorithm, it also has a favorable computational complexity. For full-graph matching, the computational complexity of the EGM, PTGM, GAGM, CGGM, and LPGM algorithms are $O(n^3),\ O(n^3),\ O(n^4),\ O(n^6)$, and $O(n^6 L)$, respectively, where L is the size of the linear programming problem. The average complexity of the SAGM algorithm is less than $O(n^6)$. Usually, $n > (r + s)^2$ so that the complexity of the RIGM algorithm is dominated by the Kuhn-

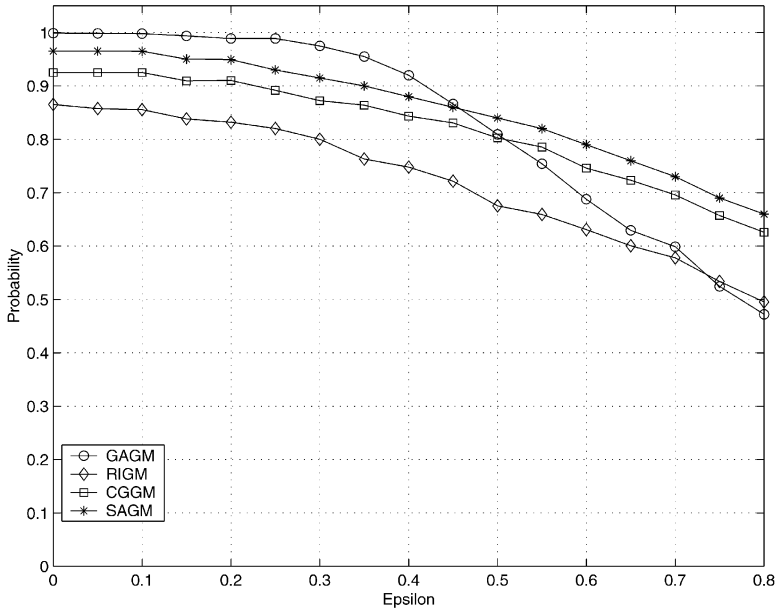

Fig. 4. Attributed subgraph matching for the case $(5/15, 5, 5)$: Probability of correct vertex-vertex matching versus ϵ.

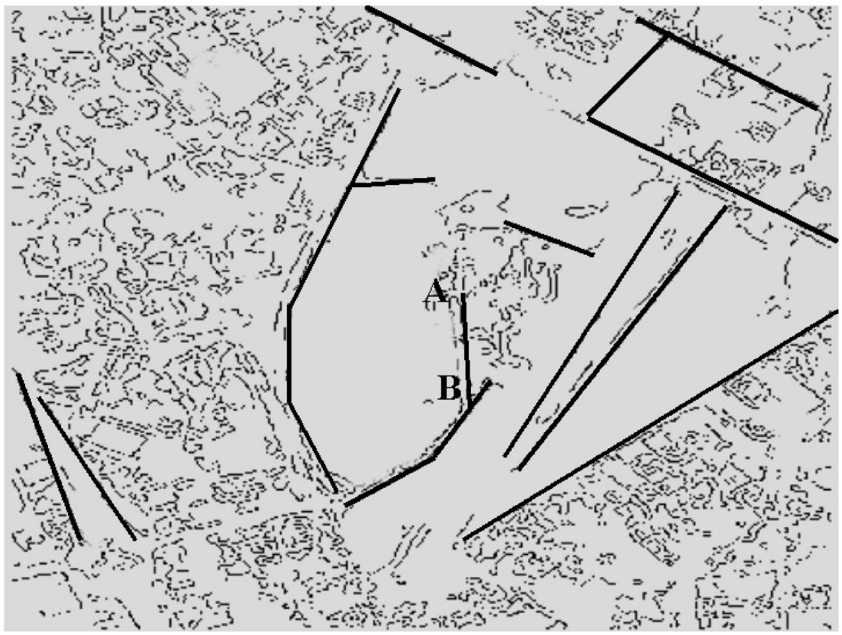

Fig. 5. Lines extracted from aerial photograph #1.

Munkres algorithm that extracts the permutation matrix from the inferred weight matrix. In this case its complexity is $O(n^3)$.

5.3 Experimental Results

For the purpose of evaluating the RIGM algorithm in a practical situation, it was applied to the problem of matching aerial photographs of airports based on the orientation of the runways and their immediate environs. Fig. 5 shows the lines extracted from an image of an airport. Fig. 6 shows the lines extracted from another image of the same airport, but viewed from a different aspect angle and direction. The bold lines superimposed on these images were extracted using a line extraction algorithm in conjunction with a neuro-fuzzy system that performed region-based processing. Only lines which lie on the borders of significant regions were used to construct the graphs. Each of the bold lines was associated with a vertex. Three features were derived for each line based on its relation to other lines and, then, used as edge attributes in the graphs. These features were based on the invariant relationships proposed by Li [50]. The vertex attributes were obtained from the neuro-fuzzy system based on a combination of features of the prominent regions adjacent to each line. The RIGM algorithm matched all lines correctly except for line A-B in Fig. 5 which was matched to line C-D in Fig. 6. From the line images, it is clear that this is the optimal match for the given graphs.

6 Conclusion

An algorithm for performing attributed full and subgraph matching was presented. The fact that the RIGM algorithm makes no assumption about the adjacency structure of the graphs to be

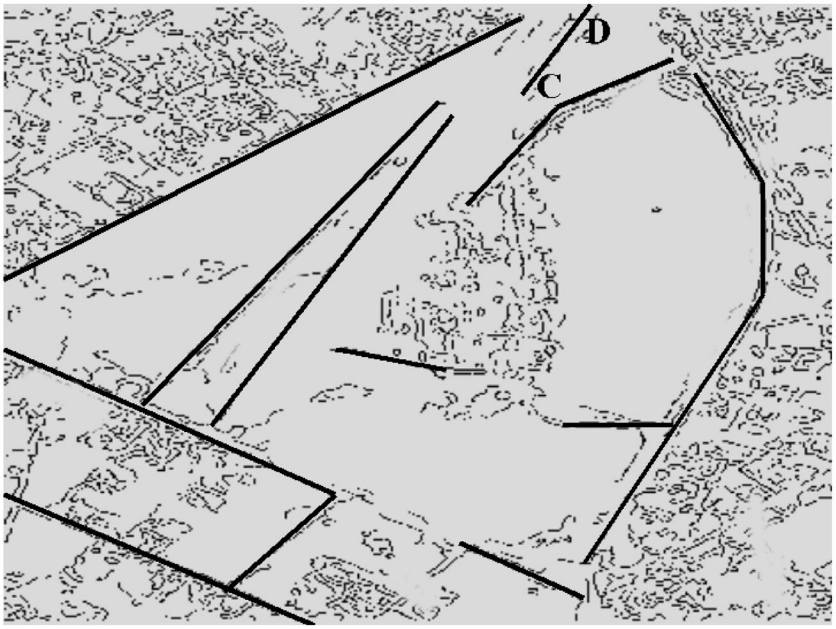

Fig. 6. Lines extracted from aerial photograph #2.

mapped and because it caters for multiple vertex and edge attributes provides an algorithm which is applicable to a vast class of matching problems. Numerical results show that RIGM algorithm performs significantly better than the LPGM, EGM, and PTGM, and is very efficient in terms of storage requirement and computational complexity, the latter being dominated by the optimal assignment problem, which it performs to produce the permutation matrix. Furthermore, the numerical results show that the RIGM algorithm performs only slightly worse that the GAGM, SAGM, and CGGM algorithms when performing subgraph matching whereas the LPGM, EGM, and PTGM algorithms are not able to perform subgraph matching.

The reduced storage requirement, low computational complexity and ease of implementation make the RIGM algorithm a suitable candidate for real-time implementations in the presence of a storage requirement constraint.

Future work include evaluation of the RIGM algorithm for different kernel functions, kernel adaptation, and the formulation of alternative interpolator constraints. The possibility of combining the RIGM algorithm with iterative relaxation techniques is also being investigated.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the US Navy under Contract No. N00014-96-11281.

REFERENCES

[1] M. A. van Wyk and T.S. Durrani, "A Framework for Multi-Scale and Hybrid RKHS-Based Approximators," *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp. 3559-3568, 2000.
[2] M.A. van Wyk and W.-H. Steeb, *Chaos in Electronics, Series: Mathematical Modeling: Theory and Applications.* vol. 2., Dordrecht, the Netherlands: Kluwer Academic, 1997.
[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
[4] M. You and K.C. Wong, "An Algorithm for Graph Optimal Isomorphism," *Proc. ICPR*, pp. 316-319, 1984.
[5] W.-H. Tsai and K.-S. Fu, "Error-Correcting Isomorphisms of Attributed Relation Graphs for Pattern Recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, pp. 757-768, 1979.
[6] W.-H. Tsai and K.-S. Fu, "Subgraph Error-Correcting Isomorphisms for Syntactic Pattern Recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 13, pp. 48-62, 1983.
[7] F. Depiero, M. Trived, and S. Serbin, "Graph Matching Using a Direct Classification of Node Attendance," *Pattern Recognition*, vol. 29, no. 6, pp. 1031-1048, 1996.
[8] M.A. Eshera and K.-S. Fu, "A Graph Distance Measure for Image Analysis," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 14, no. 3, pp. 398-408, 1984.
[9] H. Bunke and K. Shearer, "A Graph Distance Metric Based on the Maximal Common Subgraph," *Pattern Recognition Letters*, vol. 19, pp. 255-259, 1998.
[10] H. Bunke and B. Messmer, "Recent Advances in Graph Matching," *Int'l J. Pattern Recognition Artificial Intelligence*, vol. 11, no. 1, pp. 169-203, 1997.
[11] R. Allen, L. Cinque, S. Tanimoto, L. Shapiro, and D. Yasuda, "A Parallel Algorithm for Graph Matching and Its MarPlas Implementaion," *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 5, pp. 490-501, 1997.
[12] H.A.L. Almohamad, "Polynomial Transform for Matching Pairs of Weighted Graphs," *Applied Math. Modeling*, vol. 15, no. 4, pp. 216-222, 1991.
[13] H.A.L. Almohamad and S.O. Duffuaa, "A Linear Programming Approach for the Weighted Graph Matching Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 522-525, 1993.
[14] S. Umeyama, "An Eigendecomposition Approach to Weighted Graph Matching Problems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 695-703, 1988.
[15] A. Rangarajan and E. Mjolsness, "A Lagrangian Relaxation Network for Graph Matching," *IEEE Int'l Conf. Neural Networks (ICNN)*, vol. 7, pp. 4629-4634, 1994.
[16] A. D.J. Cross, C. Wilson, and E.R. Hancock, "Inexact Matching Using Genetic Search," *Pattern Recognition*, vol. 30, no. 6, pp. 953-970, 1997.
[17] D.E. van den Bout and T.K. Miller, III, "Graph Partitioning Using Annealed Neural Networks," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 192-203, 1990.
[18] C. von der Malsburg, "Pattern Recognition by Labelled Graph Matching," *Neural Networks*, vol. 1, pp. 141-148, 1988.
[19] P.D. Simic, "Constrained Nets for Graph Matching and Other Quadratic Assignment Problems," *Neural Computation*, vol. 3, pp. 268-281, 1991.
[20] E. Mjolsness and C. Garrett, "Algebraic Transformations of Objective Functions," *Neural Networks*, vol. 3, pp. 651-669, 1990.
[21] S.-S. Yu and W.-H. Tsai, "Relaxation by the Hopfield Neural Network," *Pattern Recognition*, vol. 25, no. 2, pp. 197-209, 1992.
[22] M. Peng and N. Gupta, "Invariant and Occluded Object Recognition Based on Graph Matching," *Int'l J. Electronic Eng. Education*, vol. 32, pp. 31-38, 1995.
[23] S. Shams, "Multiple Elastic Modules for Visual Pattern Recognition," *Neural Networks*, vol. 8, no. 9, pp. 1439-1465, 1995.
[24] E. Mjolsness, G. Gindi, and P. Anandan, "Optimization in Model Matching and Perceptual Organization," *Neural Computation*, vol. 1, pp. 218-229, 1989.
[25] L. Kitchen, "Relaxation Applied to Matching Quantitative Relational Structures," *IEEE Trans. Systems Man Cybernetics*, vol. 10, pp. 96-101, 1980.
[26] L. Kitchen and A. Rosenfeld, "Discrete Relaxation for Matching Relational Structures," *IEEE Trans. Systems Man Cybernetics*, vol. 9, pp. 869-874, 1979.
[27] R.C. Wilson and E.R. Hancock, "Structural Matching by Discrete Relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, pp. 634-648, 1997.
[28] L.G. Shapiro and R.M. Haraldick, "Structural Descriptions and Inexact Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, no. 6, pp. 504-519, 1981.
[29] R.A. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labelling Processes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 3, pp. 267-286, 1983.
[30] M. Pelillo and M. Refice, "Learning Compatibility Coefficients for Relaxation Labelling Processes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 933-945, 1994.
[31] L.S. Davis, "Shape Matching Using Relaxation Techniques," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 1, pp. 60-72, 1979.
[32] S. Peleg, "A New Probabilistic Relaxation Scheme," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 4, pp. 362-369, 1980.
[33] W.J. Christmas, J. Kittler, and M. Petrou, "Structural Matching in Computer Vision Using Probabilistic Relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 749-764, 1995.
[34] S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377-388, 1996.
[35] A.M. Finch, R.C. Wilson, and E.R. Hancock, "Symbolic Matching with the EM Algorithm," *Pattern Recognition*, vol. 31, no. 11, pp. 1777-1790, 1998.
[36] M.L. Williams, R.C. Wilson, and E.R. Hancock, "Multiple Graph Matching with Bayesian Inference," *Pattern Recognition Letters*, vol. 18, pp. 1275-1281, 1997.
[37] A.D.J. Cross and E.R. Hancock, "Graph Matching with a Dual Step EM Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1236-1253, 1998.
[38] R.C. Wilson and E.R. Hancock, "A Bayesian Compatibility Model for Graph Matching," *Pattern Recognition Letters*, vol. 17, pp. 263-276, 1996.
[39] B.J. van Wyk, M.A. van Wyk, and F. Virolleau, "The CGGM Algorithm and its DSP implementation," *Proc. Third European DSP Conf. Education Resources*, Sept. 2000.
[40] M.A. van Wyk and J. Clark, "An Algorithm for Approximate Least-Squares Attributed Graph Matching," *Problems in Applied Math. and Computational Intelligence*, pp. 67-72, 2001.
[41] B.J. van Wyk and M.A. van Wyk, "The Spherical Approximation Graph Matching Algorithm," *Proc. Int'l Workshop Multidisciplinary Design Optimization*, Aug., pp. 280-288 2000.
[42] M.A. van Wyk, B.J. van Wyk, and T.S. Durrani, "RKHS Interpolator-Based Graph and Subgraph Matching Algorithm," *Proc. PRASA Workshop*, Nov., 1999.
[43] M. Pelillo, K. Siddiqi, and S.W. Zucker, "Matching Hierarchical Structures Using Association Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 1105-1119, 1999.
[44] J.T. Li, K. Zhang, K. Jeong, and D. Sasha, "A System for Approximate Tree Matching," *IEEE Trans. Knowledge and Data Eng.*, vol. 6, pp. 559-571, 1994.
[45] S.Y. Lu, "A Tree Matching Algorithm Based on Tree Merging and Node Splitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 249-255, 1984.
[46] D.W. Matula, "An Algorithm for Subtree Identification," *SIAM Rev.*, vol. 10, pp. 273-274, 1968.
[47] S.W. Reyner, "An Analysis of a Good Algorithm for the Subtree Problem," *SIAM J. Computing*, vol. 6, pp. 730-732, 1977.
[48] D. Sasha, J.T.L. Wang, K. Zhang, and F.Y. Shih, "Exact and Approximate Algorithms for Unordered Tree Matching," *IEEE Trans. Systems. Man, and Cybernetics*, vol. 24, pp. 668-678, 1994.
[49] P.R. Halmos, *Measure Theory.* Princeton, NJ: Van Nostrand Co., 1964.
[50] S.Z. Li, "Matching: Invariant to Translations Rotations and Scale Changes," *Pattern Recognition*, vol. 25, no. 8, pp. 583-594, 1992.
[51] R.J.P. De Figueiredo and G. Chen, *Nonlinear Feedback Control Systems: An Operator Theory Approach.* San Diego, CA: Academic Press, 1993.
[52] S. Haykin, *Adaptive Filter Theory.* Upper Saddle River, NJ: Prentice Hall, 1996.
[53] D.G. Luenberger, *Optimization by Vector Space Methods.* New York, NY: John Wiley & Sons, 1969.
[54] B. Noble and J.W. Daniel, *Applied Linear Algebra, Third ed.* Englewood Cliffs, NJ: Prentice Hall, 1988.
[55] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Englewood Cliffs, NJ: Prentice-Hall, 1982.
[56] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing.* New York, NY: Wiley, 1993.