Attributed Tree Homomorphism Using Association Graphs

Massimo Bartoli **Marcello Pelillo**
Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino 155, 30172 Venezia Mestre, Italy
{mbartoli,pelillo}@dsi.unive.it

Kaleem Siddiqi
Center for Intelligent Machines
McGill University
3480 University Street, Montréal, PQ, H3A 2A7 Canada
siddiqi@cim.mcgill.ca

Steven W. Zucker
Center for Computational Vision and Control
Yale University
P.O. Box 208285, New Haven CT 06520-8285
zucker-steven@cs.yale.edu

Abstract

The matching of hierarchical relational structures is of significant interest in computer vision and pattern recognition. We have recently introduced a new solution to this problem, based on a maximal clique formulation in a (derived) "association graph." This allows us to exploit the full arsenal of clique finding algorithms developed in the algorithms community. However, thus far we have focussed on one-to-one correspondences (isomorphisms), which appears to be too strict a requirement for many vision problems. In this paper we provide a generalization of the association graph framework to handle many-to-one correspondences. We define a notion of an ϵ-homomorphism (a many-to-one mapping) between attributed trees, and provide a method of constructing an association graph where maximal cliques are in one-to-one correspondence with maximal subtree homomorphisms. We then solve the problem by using *replicator* dynamical systems from evolutionary game theory. An extension of the framework takes similarity between attribute vectors into account, and is shown to be equivalent to the maximum weight clique problem. We illustrate the approach by matching articulated and deformed shapes described by shock trees.

1 Introduction

The matching of relational structures is a classic problem in computer vision and pattern recognition, instances of which arise in areas as diverse as object recognition, motion and stereo analysis (see, e.g., [2, 10]). A well known approach to solving this problem consists of transforming it into an equivalent problem of finding a maximum clique in an auxiliary graph structure, known as the *association graph* [2, 3]. The idea goes back to Ambler *et al.* [1], and has since been successfully employed in a variety of different problems, e.g., [5, 21, 12, 25, 20]. This framework is attractive because it casts relational structure matching as a pure graph-theoretic problem, for which a solid theory and powerful algorithms have been developed. Although the maximum clique problem is known to be NP-complete [9], powerful heuristics have been developed which efficiently find good approximate solutions [16].

In many computer vision problems, relational structures are organized in a hierarchical manner, i.e., are *trees* (see, for example, [15, 22, 23, 14]). However, in standard association graph formulations, the solutions are not constrained to preserve this partial order. Hence, the extension of such techniques to tree matching problems is of considerable interest. We have recently introduced a solution to this problem by providing a novel way of deriving an association graph, based on the graph-theoretic notions of connectivity and the distance matrix [19]. We have proved that in this new formulation there is a one-to-one correspondence between maximum cliques in the derived association graph, and maximum subtree isomorphisms. The framework has also been extended to handle the matching of trees whose nodes have one or more associated attributes, by casting the attributed tree matching problem as an equivalent problem of finding a maximum weight clique in a "weighted" association graph.

Whereas thus far we have focussed on one-to-one relations (isomorphisms) between the structures being matched, it is clear that this is too strict a requirement for problems where structural alterations may occur in the underlying trees, i.e., nodes are deleted or additional nodes are present. In this paper we provide a generalization of the association graph framework to handle many-to-one correspondences. We define a notion of an ϵ-homomorphism (a many-to-one mapping) between attributed trees, and provide a method of constructing an association graph where maximal cliques are in one-to-one correspondence with maximal subtree homomorphisms. We then solve the problem by using *replicator* dynamical systems from evolutionary game theory. A extension of the framework takes similarity between attribute vectors into account, and is shown to be equivalent to the maximum weight clique problem. We illustrate the approach by matching articulated and deformed shapes described by shock trees.

2 Tree homomorphisms and maximal cliques

2.1 Notations and definitions

Before going into the details of the proposed framework, we need to introduce some graph-theoretic notations and definitions. More details can be found in standard textbooks of graph theory, such as [11]. Let $G = (V, E)$ be a graph, where V is the set of nodes and E is the set of (undirected) edges. The *order* of G is the number of nodes in V, while its *size* is the number of edges. Two nodes $u, v \in V$ are said to be *adjacent* (denoted $u \sim v$) if they are connected by an edge. A *path* is any sequence of distinct nodes $u_0 u_1 \dots u_n$ such that for all $i = 1 \dots n$, $u_{i-1} \sim u_i$; in this case, the *length* of the path is n. If $u_0 = u_n$ the path is called a *cycle*. A graph is said to be *connected* if any pair of nodes is joined by a path. The *distance* between two nodes u and v, denoted by $d(u, v)$, is the length of the shortest path joining them (by convention $d(u, v) = \infty$, if there is no such path). We extend the notion of adjacency to subsets of nodes of G. Two disjoint subsets of nodes C and D are said to be adjacent (denoted $C \sim D$) if $\min_{u \in C,\ v \in D} d(u, v) = 1$. It is clear that when C and D are singletons, this is equivalent to the traditional notion of adjacency between nodes. We shall therefore not make any distinction between the two concepts in what follows. Given a subset of nodes $C \subseteq V$, the *induced subgraph* $G[C]$ is the graph having C as its node set, and two nodes are adjacent in $G[C]$ if and only if they are adjacent in G.

A connected graph with no cycles is called a *tree*. A *rooted* tree is one which has a distinguished node, called the *root*. The *level* of a node u in a rooted tree, denoted by $\mathrm{lev}(u)$, is the length of the path connecting the root to u. Note that there is an obvious equivalence between rooted trees and directed trees, where the edges are assumed to be oriented. We shall therefore use the same terminology typically used for directed trees to define the relation between two adjacent nodes. In particular, if $u \sim v$ and $\mathrm{lev}(v) - \mathrm{lev}(u) = +1$, we say that u is the *parent* of v and, conversely, v is a *child* of u. Trees have a number of interesting properties. One which turns out to be very useful for our characterization is that in a tree any two nodes are connected by a *unique* path.

In [19] we introduced the notion of a *path-string*, which turned out to be central to the development of our tree isomorphism framework. Let u and v be two distinct nodes of a rooted tree T, and let $u = x_0 x_1 \dots x_n = v$ be the (unique) path joining them. The *path-string* of u and v, denoted by $\mathrm{str}(u, v)$, is the string $s_1 s_2 \dots s_n$ on the alphabet $\{-1, +1\}$ where, for all $i = 1 \dots n$, $s_i = \mathrm{lev}(x_i) - \mathrm{lev}(x_{i-1})$.

The path-string concept has a very intuitive meaning. Suppose that you stand on a particular node in a rooted tree and want to move to another adjacent node. Because of the orientation induced by the root, only two types of moves can be done, i.e., going down to one of the children (if one exists) or going up to the parent (if you are not on the root). Let us assign to the first move the label $+1$, and to the second the label -1. Now, suppose that you want to move from node u to v, following the unique path joining them. Then, the path-string of u and v is simply the string of elementary moves required to reach v, starting from u. It may be thought of as the degree of relationship between two relatives in a "family" tree.

Observe that path-strings cannot be arbitrary strings of -1's and $+1$'s. Since trees do not have cycles, in fact, once we go down one level along a path (i.e., we make a "$+1$" move) we cannot return to the parent. This is formally stated by saying that if $\mathrm{str}(u, v) = s_1 s_2 \dots s_n$ is the path-string between any two nodes u and v, then $s_i = +1$ implies $s_j = +1$ for all $j \geq i$.

We define the *path-pair* of any two nodes u and v in a tree as $\mathrm{pair}(u, v) = (n, p)$, where n is the number of negative components in $\mathrm{str}(u, v)$ and p is the number of positive components in $\mathrm{str}(u, v)$. It is clear from the previous observation that path-pairs and path-strings are equivalent concepts.

Formally, an *attributed tree* is a triple $T = (V, E, \alpha)$, where (V, E) is the "underlying" rooted tree and $\alpha : V \to \mathcal{A}$ is a function which assigns an attribute vector $\alpha(u)$ to each node $u \in V$. We shall also consider a function $\delta : \mathcal{A} \to \Re^*_+$ which assigns to each set of attributes (and therefore to each node in the tree) a real positive number. This function will be interpreted as the negligibility of the nodes in the tree. Specifically, a node will be declared "negligible" if the value of the function δ corresponding to its attributes is smaller than a fixed threshold ϵ. This will allow us to associate a *cluster* of nodes in the first subtree to a single node in the other one, thereby defining a many-to-one mapping from the first to the second tree. Fig. 1 shows an example of a many-to-one matching between trees.

For technical simplicity, we shall transform our node-weighted tree into an edge-weighted

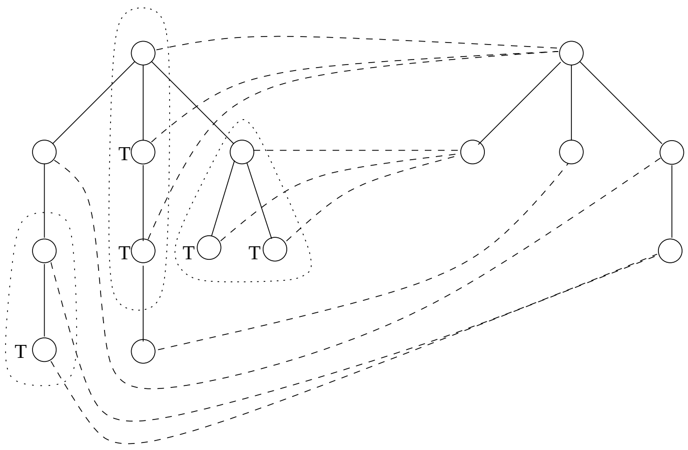


Figure 1: An example of a many-to-one matching between trees. The label "T" denotes negligible nodes.

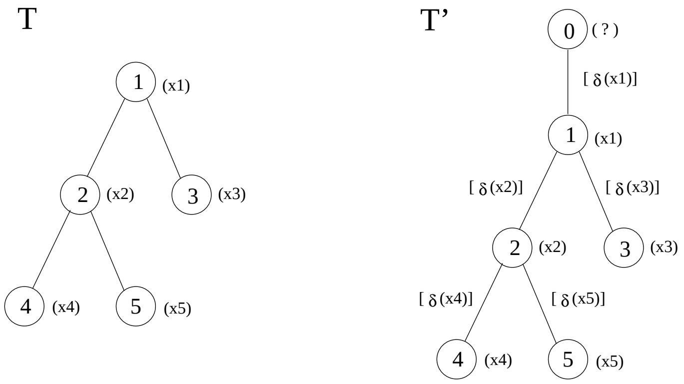


Figure 2: An example of a derived attributed tree. Attribute vectors associated with each node are indicated by round brackets, and edge weights by square brackets.

derived attributed tree, by simply moving the δ-value associated to every node to the edge connecting it to its parent. The root of the original tree will become a child of a newly created dummy root, and its weight will be moved to the corresponding edge. In the derived tree, we shall speak of a "negligible" edge, when its weight is less than a fixed threshold ϵ. An example of a derived attributed tree is shown in Fig. 2.

Given a fixed threshold $\epsilon \geq 0$, we define the distance $d_\epsilon(u, v)$ between two nodes u and v in a tree, as the number of non-negligible edges on the unique path from u to v. Given a subset of nodes $C \subseteq V$, $lev_\epsilon(C)$ is the distance d_ϵ between the node in C nearest to the root and the root itself.

2.2 Deriving the association graph

Let $T_1(V_1, E_1, \alpha_1)$ and $T_2(V_2, E_2, \alpha_2)$ be two attributed trees. Any function $\gamma : H_1 \rightarrow H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is called a *subtree ϵ-homomorphism* if the induced subgraphs $T_1[H_1]$ and $T_2[H_2]$ are trees and if for any $u, v \in T_2$ it satisfies the following properties:

$$\forall w, z \in \gamma^{-1}(u),\ d_\epsilon(w, z) = 0 \tag{1}$$

$$u \sim v \Leftrightarrow \gamma^{-1}(u) \sim \gamma^{-1}(v) \tag{2}$$

$$lev(u) - lev(v) = -1 \Leftrightarrow lev_\epsilon(\gamma^{-1}(u)) - lev_\epsilon(\gamma^{-1}(v)) \in \{-1, 0\} \tag{3}$$

Intuitively, property (1) imposes that inside any cluster in T_1, all pair of nodes are connected by negligible edges. Property (2) preserves topological order between nodes, while property (3) preserves hierarchical relations. A subtree ϵ-homomorphism is *maximal* if there is no other subtree ϵ-homomorphism $\gamma' : H_1' \to H_2'$ with H_1 a strict subset of H_1', and *maximum* if H_1 has largest cardinality.

We now introduce the notion of an *ϵ-path-pair* which is a generalization of the path-pair/path-string concept.

Definition 1 *Let u and v be two nodes of an attributed tree, and let $\epsilon \geq 0$ be a fixed threshold. The* ϵ-path-pair *of u and v, denoted by ϵ-pair(u, v), is the quadruple $(n, n^{(\epsilon)}, p, p^{(\epsilon)})$, where n and p are the number of positive and negative components of the path-string of u and v, respectively, $n^{(\epsilon)}$ is the number of negative path-string entries corresponding to negligible edges (i.e., with weight not exceeding ϵ), and $p^{(\epsilon)}$ is the number of positive entries corresponding to negligible edges in the derived attributed tree. Clearly, $n^{(\epsilon)} \leq n$ and $p^{(\epsilon)} \leq p$.*

Observe that, when $\epsilon = 0$, the ϵ-path-pair becomes equivalent to the notion of a path-string (or a path-pair).

Now we define the association graph.

Definition 2 *The ϵ-tree association graph (ϵ-TAG) of two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ is the graph $G_\epsilon(V, E)$ where*

$$V = V_1 \times V_2$$

and for any two nodes (w, u) and (z, v) in V, we have:

$$(w, u) \sim (z, v) \Leftrightarrow (0 \leq n_1 - n_2 \leq n_1^{(\epsilon)}) \wedge (0 \leq p_1 - p_2 \leq p_1^{(\epsilon)}) \ , \tag{4}$$

where ϵ-pair$(w, z) = (n_1, n_1^{(\epsilon)}, p_1, p_1^{(\epsilon)})$, and ϵ-pair$(u, v) = (n_2, n_2^{(\epsilon)}, p_2, p_2^{(\epsilon)})$.

Intuitively, condition (4) imposes that the number of ascent edges from node w to node z must be greater than or equal to the number of ascent edges between the nodes u and v and, additionally, the number of negligible ascent edges between w and z must be less than or equal to the number of ascent edges between u and v. The same applies to the descent edges. Notice that, when $\epsilon = 0$ we have $n_1^{(\epsilon)} = n_2^{(\epsilon)} = p_1^{(\epsilon)} = p_2^{(\epsilon)} = 0$, and therefore we obtain the same association graph structure as originally studied for the tree isomorphism case [19]. A subset of vertices of G is said to be a *clique* if all its nodes are mutually adjacent. A *maximal* clique is one which is not contained in any larger clique, while a *maximum* clique is a clique having largest cardinality. The maximum clique problem is to find a maximum clique of G. The following theorem, which is the basis of the work reported here, establishes a one-to-one correspondence between the maximum subtree ϵ-homomorphism problem and the maximum clique problem (see [4] for a complete proof).

Theorem 1 *Any maximal (maximum) subtree ϵ-homomorphism between two rooted attributes trees induces a maximal (maximum) clique in the corresponding ϵ-TAG, and vice versa.*

Proof (outline). Let $\gamma : H_1 \to H_2$ be a maximal ϵ-homomorphism between two weighted trees T_1 and T_2 and let $G_\epsilon = (V, E)$ be the corresponding ϵ-TAG. Let $C_\gamma \subseteq V$ be defined as:

$$C_\gamma = \{(w, u) : u \in \gamma(H_1), w \in \gamma^{-1}(u)\} \ .$$

From the definition of ϵ-homomorphism it follows that γ maps the path between any two nodes $u, v \in \gamma(H_1)$ onto the set of paths joining the nodes of $\gamma^{-1}(u)$ with the nodes of $\gamma^{-1}(v)$. This implies that if ϵ-pair$(u, v) = (n_1, 0, p_1, 0)$ and ϵ-pair$(w, z) = (n_2, n_2^{(\epsilon)}, p_2, p_2^{(\epsilon)})$ with $w \in \gamma^{-1}(u)$ and $z \in \gamma^{-1}(v)$, then $(0 \leq n_2 - n_1 \leq n_2^{(\epsilon)}) \wedge (0 \leq p_2 - p_1 \leq p_2^{(\epsilon)})$ for all $u, v \in \gamma(H_1)$ and therefore C_γ is a clique. Trivially C_γ is a maximal clique because γ is maximal, and this proves the first part of the theorem.

Suppose now that $C = \{(w_{11}, u_1), (w_{12}, u_1), \ldots, (w_{1n}, u_1), \ldots, (w_{m1}, u_m), \ldots, (w_{mt}, u_m)\}$ is a maximal clique of G_ϵ, and let $H_1 = \{w_{11}, w_{12}, \ldots, w_{1n}, \ldots, w_{m1}, \ldots, w_{mt}\} \subseteq V_1$ and $H_2 = \{u_1, u_2, \ldots, u_m\} \subseteq V_2$. Define $\gamma : H_1 \to H_2$ as $\gamma(w_{ij}) = u_i$, for all $i = 1 \ldots m$. From the definition of the ϵ-TAG and the hypothesis that C is a clique, it is simple to see that γ is a many-to-one and onto correspondence between H_1 and H_2. It is also possible to show that γ enjoys the properties (1)–(3) of ϵ-homomorphisms [4]. The fact that γ is a maximal homomorphism is a straightforward consequence of the maximality of C.

To conclude the proof we have to show that the induced subgraphs $T_1[H_1]$ and $T_2[H_2]$ are trees, and this is equivalent to showing that they are connected. The basic idea behind the proof is to suppose by contradiction that this is not the case, and to find a node x^* in T_2 and a node y^* in T_1 such that $\{(y^*, x^*)\} \cup C \subseteq V$ is a clique.

Let us start by assuming that $T_1[H_1]$ is disconnected, and let $y_p \notin H_1$ be any node on a path between two nodes in H_1 which is not completely contained in H_1. Moreover, let w_{is} and w_{jt} be two nodes in H_1 closest to y_p. Three different cases may arise, and we shall consider them separately. First, if $\gamma(w_{is}) = \gamma(w_{jt})$ then we let $x^* = \gamma(w_{is})$ and $y^* = y_p$. If $\gamma(w_{is}) \sim \gamma(w_{jt})$ then we let $y^* = y_p$, and choose x^* as either $\gamma(w_{is})$ or $\gamma(w_{jt})$ depending on the type of edges connecting y_p to w_{is} and w_{jt}. Finally, in the more general case when $\gamma(w_{is})$ and $\gamma(w_{jt})$ are not adjacent, it can be proven that for any node x in the $\gamma(w_{is})\gamma(w_{jt})$-path there exists a y^* in the $w_{is}w_{jt}$-path such that in G_ϵ node (y^*, x) is adjacent to both (w_{is}, u_i) and (w_{jt}, u_j).

It can be seen [4] that in all the previous cases the set $\{(y^*, x^*)\} \cup C \subseteq V$ is a clique, and this contradicts the hypothesis that C is a maximal clique. This can be proved by exploiting the obvious fact that if x is a node on the path joining any two nodes u and v, then ϵ-pair$(u, v) = \epsilon$-pair$(u, x) + \epsilon$-pair(x, v), where "+" denotes the usual sum between vectors. Similar arguments apply to the case when $T_2[H_2]$ is assumed to be disconnected and we omit the details.

The "maximum" part of the statement is proved similarly. $\diamond$

Once the tree homomorphism problem has been formulated as a maximum clique problem, any clique finding algorithm can be employed to solve it (see [6] for a recent review). In the work reported in this paper, we used an approach recently introduced in [7], which consists of transforming the maximum clique problem to an equivalent quadratic program via the Motzkin-Straus theorem. We then used a class of dynamical systems from evolutionary game theory called *replicator equations* to find approximate solutions of it. Due space limitations, we refer the reader to [18, 19].

3 An example: Matching shock trees

We illustrate our framework with numerical examples of shape matching. We use a *shock graph* representation based on a coloring of the shocks (singularities) of a curve evolution process acting on simple closed curves in the plane [13]. Shocks are grouped into distinct types according to the local variation of the radius function along the medial axis. Intuitively, the radius function varies monotonically at a type 1, reaches a strict local minimum at a type 2, is constant at a type 3 and reaches a strict local maximum at a type 4. The shocks comprise vertices in the graph, and their formation times direct edges to form a basis for tree matching; see [23] for details. Each graph can be reduced to a unique attributed rooted tree, providing the requisite hierarchical structure for our matching algorithm. The vector of attributes assigned to each node $u \in V$ of the attributed shock tree $T = (V, E, \alpha)$ is given by $\alpha(u) = (x_1, y_1, r_1, v_1, \theta_1; ...; x_m, y_m, r_m, v_m, \theta_m)$. Here m is the number of shocks in the group, and $x_i, y_i, r_i, v_i, \theta_i$ are, respectively, the x coordinate, the y coordinate, the radius (or time of formation), the speed, and the direction of each shock i in the sequence, obtained as outputs of the shock detection process [24].

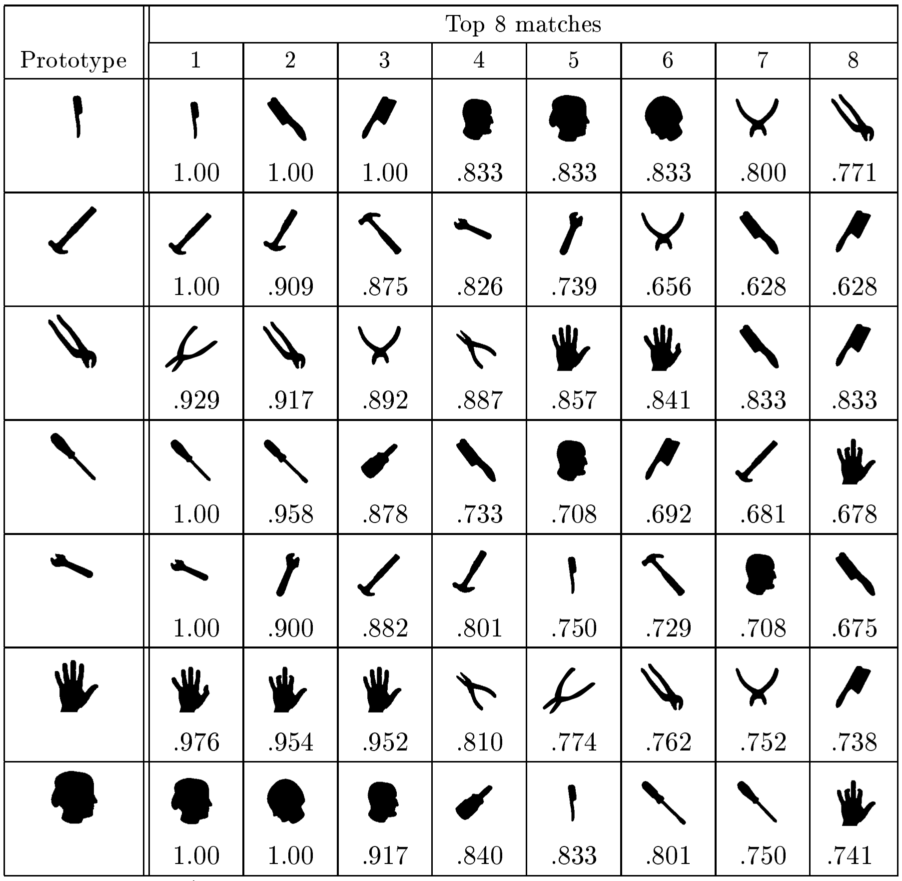
We selected 21 silhouettes representing eight different object classes (bottom row of Fig. 5); the tools shapes were taken from the Rutgers Tools database. For each object class a prototype object was chosen and matched against *all* entries in the database using the clique-finding replicator equations described in [17, 19, 18].

The selection of a proper value for the parameter ϵ is clearly crucial for the performance of the matching process. Figure 3 illustrates the behavior of the algorithm as a function of ϵ, on two synthetic trees. When $\epsilon = 0$ we have the one-to-one matching case. The overall emerging picture (which is consistent with our intuition) is that the larger the value of ϵ, the larger the cluster formed by the matching process. In the experiments presented here we set $\epsilon = 0.002$, a value which we found to be near-optimal for the trees at hand. Figure 4 shows two representative matching examples on shock trees, one with $\epsilon = 0$ (i.e., isomorphism) and the other with $\epsilon = 0.002$. It can be seen that a cluster of negligible nodes is formed and matched against a single node of the second tree.

For each pair of matched trees the following score was computed:

$$I = \frac{size_1 \cdot n_2 + size_2 \cdot n_1}{2 \cdot n_1 \cdot n_2} \tag{5}$$

where $size_1$ and $size_2$ are the sizes of the first and the second subtrees put into correspondence, and n_1 and n_2 are the sizes of the trees being matched. This score takes values in the range $[0, 1]$, and if $size_1 = size_2$, I coincides with the measure used in [19]. The top 8 matches for each query shape, along with the associated scores, are shown in Table 1. Note that despite the fact that metric/label information associated with nodes in the shock trees was discounted altogether, all exemplars in the same class as the query shape are within the top matches. These results represent an improvement with respect to those obtained for the isomorphism case (one-to-one) [19], which corresponds to having $\epsilon = 0$, since nodes with negligible significance can now be clustered. Specifically, we observed that our many-to-one matching algorithm tends to provide a sharper distinction between object classes. This is illustrated in Figure 5, where the scores obtained in matching an example prototype object (a brush) against all other objects in the database is reported for both $\epsilon = 0$ (isomorphism) and $\epsilon = 0.002$ (homomorphism).



Prototype	Top 8 matches							
	1	2	3	4	5	6	7	8
	1.00	1.00	1.00	.833	.833	.833	.800	.771
	1.00	.909	.875	.826	.739	.656	.628	.628
	.929	.917	.892	.887	.857	.841	.833	.833
	1.00	.958	.878	.733	.708	.692	.681	.678
	1.00	.900	.882	.801	.750	.729	.708	.675
	.976	.954	.952	.810	.774	.762	.752	.738
	1.00	1.00	.917	.840	.833	.801	.750	.741

Table 1: A tabulation of the top 8 matches for each prototype shape, with $\epsilon = 0.002$. The scores indicate the value of index (5).

4 Attributed tree matching and weighted TAG's

In the previous section, node attributes have only been used to induce clusters of nodes in the many-to-one matching process. Clearly, in realistic applications, it would be desirable to find a homomorphism which pairs nodes having "similar" attributes. To this end, let σ be any similarity measure on the attribute space, i.e., any (symmetric) function which assigns a positive number to any pair of attribute vectors. If $\gamma : H_1 \to H_2$ is a subtree ϵ-homomorphism between two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$, the overall similarity between the induced subtrees $T_1[H_1]$ and $T_2[H_2]$ can be defined as follows:

$$S(\gamma) = \sum_{u \in \gamma(H_1)} \sum_{w \in \gamma^{-1}(u)} \sigma(\alpha_1(w), \alpha_2(u))$$

The ϵ-homomorphism γ is called a *maximal similarity subtree ϵ-homomorphism* if there is no other subtree ϵ-homomorphism $\gamma' : H_1' \to H_2'$ such that H_1 is a strict subset of H_1' and $S(\gamma) < S(\gamma')$. It is called a *maximum similarity subtree ϵ-homomorphism* if $S(\gamma)$ is largest among all subtree ϵ-homomorphisms between T_1 and T_2.

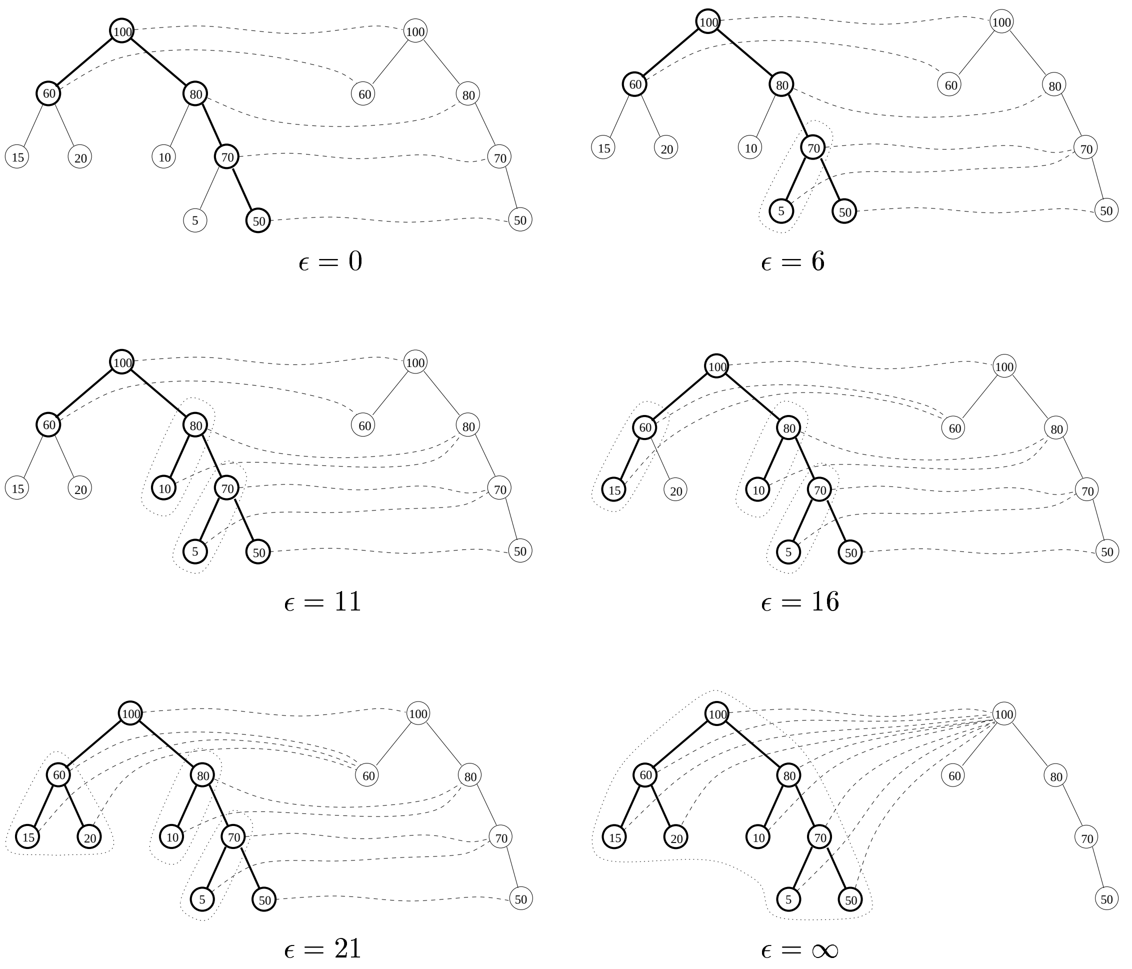


Figure 3: The behavior of the matching algorithm on two example trees, as a function of the parameter ϵ.

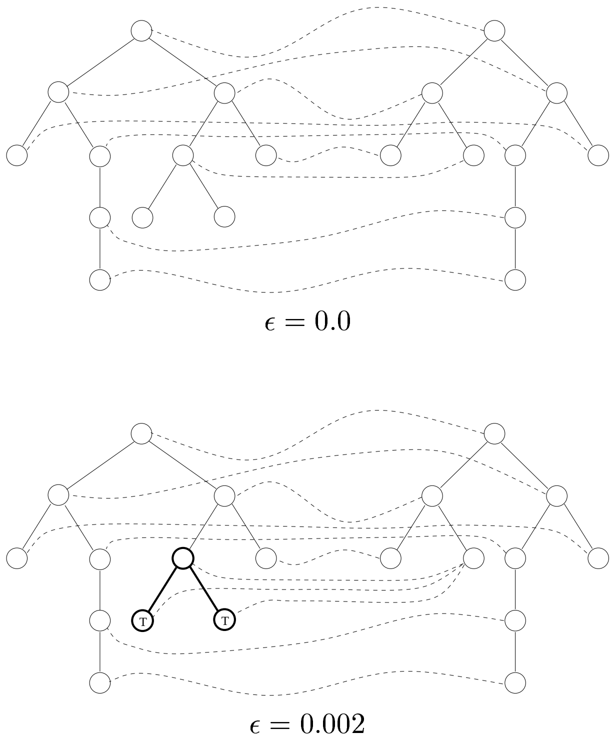


Figure 4: An example illustrating the formation of clusters in the many-to-one matching process. The nodes labeled with a "T" represent negligible nodes.

The *weighted ϵ-TAG* of two attributed trees T_1 and T_2 is the weighted graph $G_\epsilon = (V, E, \omega)$ where V and E are defined as in Definition 2, and ω is a function which assigns a positive weight to each node $(u, v) \in V = V_1 \times V_2$ as follows:

$$\omega(u, v) = \sigma(\alpha_1(u), \alpha_2(v)) \ .$$

Given a subset of nodes C of V, the total weight assigned to C is simply the sum of all the weights associated with its nodes. A *maximal weight* clique in G is one which is not contained in any other clique having larger total weight, while a *maximum weight* clique is a clique having largest total weight. The maximum weight clique problem is to find a maximum weight clique of G [6]. Note that the unweighted version of the maximum clique problem arises as a special case when all the nodes are assigned a constant weight.

The following result, which is the weighted counterpart of Theorem 1, establishes a one-to-one correspondence between the attributed tree matching problem and the maximum weight clique problem (the proof is essentially identical to that of Theorem 1).

Theorem 2 *Any maximal (maximum) similarity subtree ϵ-homomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted ϵ-TAG, and vice versa.*

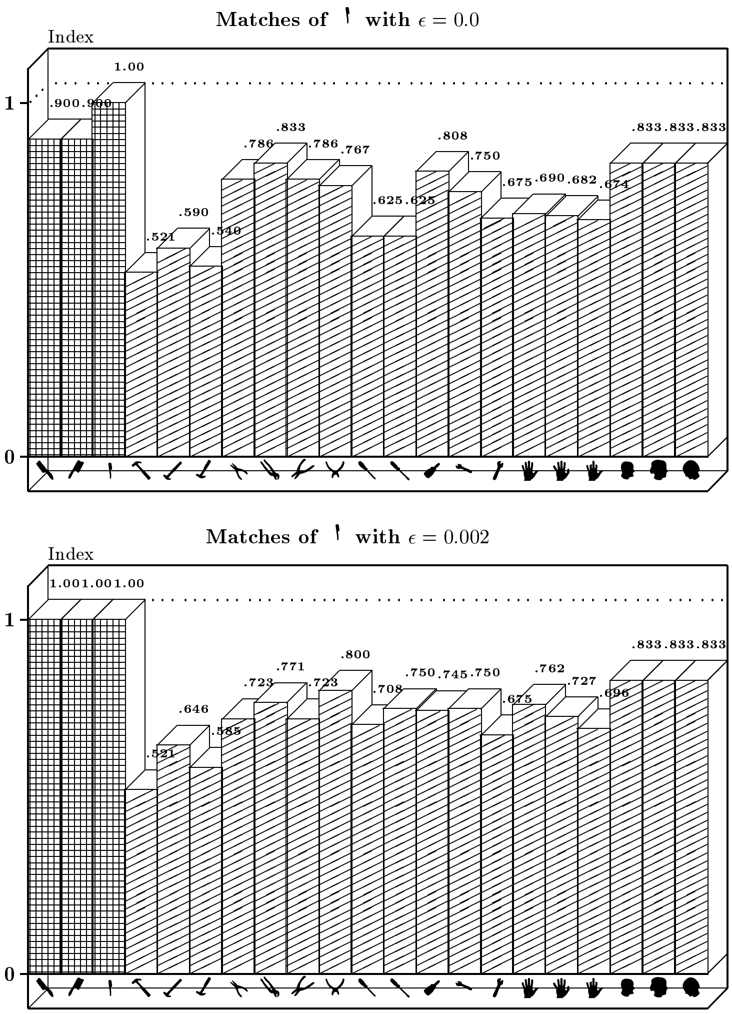


Figure 5: The scores obtained from matching an example prototype object against all other objects in the database, for $\epsilon = 0$ and $\epsilon = 0.002$.

We now provide examples of weighted shock tree matching, using the geometric attributes associated with shock tree nodes. In order to apply our framework we must define a similarity measure between the attributes of two nodes u and v. The similarity measure we use is a linear combination of four terms, incorporating the differences in lengths, radii, velocities, and curvature of the two shock sequences, respectively. Each term is normalized to provide a unitless quantity, so that these different geometric properties can be combined. Let u contain m shocks and v contain n shocks, and without loss of generality, assume that $m \geq n$. The Euclidean length of each sequence of shocks is given by:

$$L(u) = \sum_{i=1}^{m-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}, \quad L(v) = \sum_{j=1}^{n-1} \sqrt{(x_j - x_{j+1})^2 + (y_j - y_{j+1})^2}.$$

Let $\xi(i) : i \rightarrow \lceil \frac{i.n}{m} \rceil$ be the many-to-one mapping of each index $i \in \{1, ..., m\}$ to an index $j \in \{1, ..., n\}$. The similarity measure between the two attribute vectors used in our experiments is defined as

$$\sigma(\alpha(u), \alpha(v)) = 1 - \beta_l \frac{|L(u) - L(v)|}{max(L(u), L(v))} - \beta_r \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{r(i) - r(\xi(i))}{max(r(i), r(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}}$$
$$-\beta_v \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{v(i) - v(\xi(i))}{max(v(i), v(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}} - \beta_{\Delta\theta} \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\Delta\theta(i) - \Delta\theta(\xi(i))}{max(\Delta\theta(i), \Delta\theta(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}} \quad (6)$$

where $\beta_l, \beta_r, \beta_v, \beta_{\Delta\theta}$ are nonnegative constants summing to 1, and $\Delta\theta$ is the change in orientation at each shock. The measure provides a number between 0 and 1, which represents the overall similarity between the geometric attributes of the two nodes being compared. The measure is designed to be invariant under rotations and translations of two shapes, and to satisfy the requirements of the weight function discussed above.

We repeated the earlier experiments, but now with weights $w(u, v) = \sigma(\alpha(u), \alpha(v))$ placed on each node (u, v) of a weighted TAG. The weights were defined by (6), with $\beta_l, \beta_r, \beta_v, \beta_{\Delta\theta}$ set to 0.25,0.4,0.2 and 0.15, respectively. We verified that the overall results were not sensitive to the precise choice of these parameters, or to slight variations in the similarity measure such as the use of a different norm. We ranked the results using a score given by the quantity

$$W \times \frac{1}{2}\{ \sum_{u \in H_1} (m(u)/M_1) + \sum_{v \in H_2} (m(v)/M_2)\}, \quad (7)$$

where H_1 and H_2 are the sets of matched nodes in the first and in the second tree, respectively, W is the overall similarity between matched nodes (i.e., the weight of the maximal clique found), M_1 and M_2 the total mass associated with the nodes of each tree, and $m(u), m(v)$ the masses of nodes u and v, respectively. The score represents the weight of the maximal clique scaled by the average of the total (relative) mass of nodes in each tree that participates in the match. As before, the top 8 matches are shown for each query shape, in Table 2. It is evident that for almost all queries, performance improves with the incorporation of geometric attributes, with better overall discrimination between the shapes. Figure 6 depicts the scores obtained when matching a prototype brush against all other shapes in the database, for both $\epsilon = 0$ (isomorphism) and $\epsilon = 0.0026$ (homomorphism).

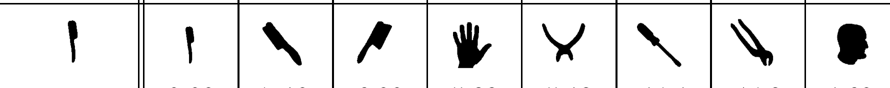
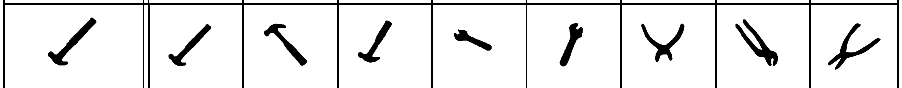
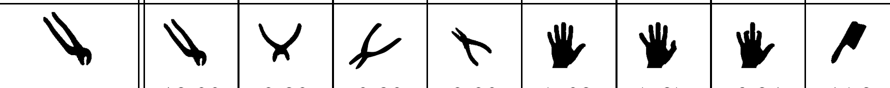
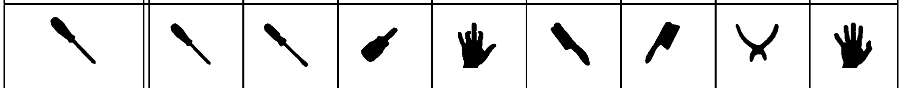
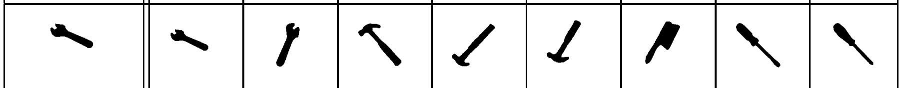
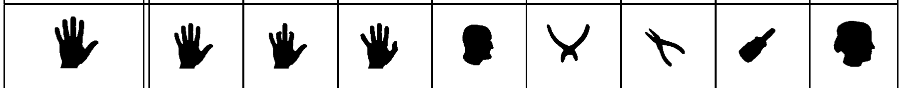
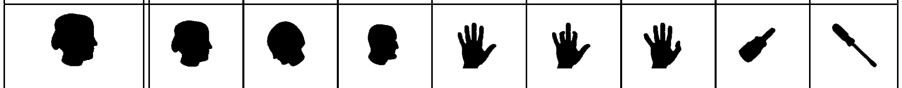
Prototype	Top 8 matches 1	2	3	4	5	6	7	8
	8.00	7.18	6.89	5.23	5.12	4.74	4.73	4.68
	9.00	7.61	6.76	5.33	4.99	3.89	3.68	3.67
	12.00	8.29	8.28	8.08	7.92	7.27	6.24	4.73
	12.00	8.96	6.35	4.57	4.37	4.28	3.83	3.81
	8.00	5.57	5.50	5.30	5.00	3.95	3.65	3.30
	21.00	19.62	18.98	7.38	6.84	6.30	6.17	6.06
	12.00	9.67	7.38	6.40	6.23	5.95	5.10	3.67

Table 2: A tabulation of the top 8 matches for each prototypes with $\epsilon = 0.0026$ (weighted case). The scores indicate the value of index (7). Compare with Table 1.

5 Conclusions

We have expanded our earlier work on matching hierarchical relational structures to handle many-to-one mappings. This has been done by introducing a notion of ϵ-homomorphism between attributed trees, and by constructing an association graph whose maximal cliques are shown to be in one-to-one correspondence with maximal subtree homomorphisms. An extension of the framework takes similarity between attribute vectors into account, and is shown to be equivalent to the maximum weight clique problem. Computational examples of matching articulated and deformed shapes give improved results over our earlier work [19], because the framework now allows negligible nodes, which arise due to structural alterations in the underlying shapes, to be ignored during the matching process.

References

[1] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. A versatile computer-controlled assembly system. In *Proc. 3rd Int. J. Conf. Artif. Intell.*,

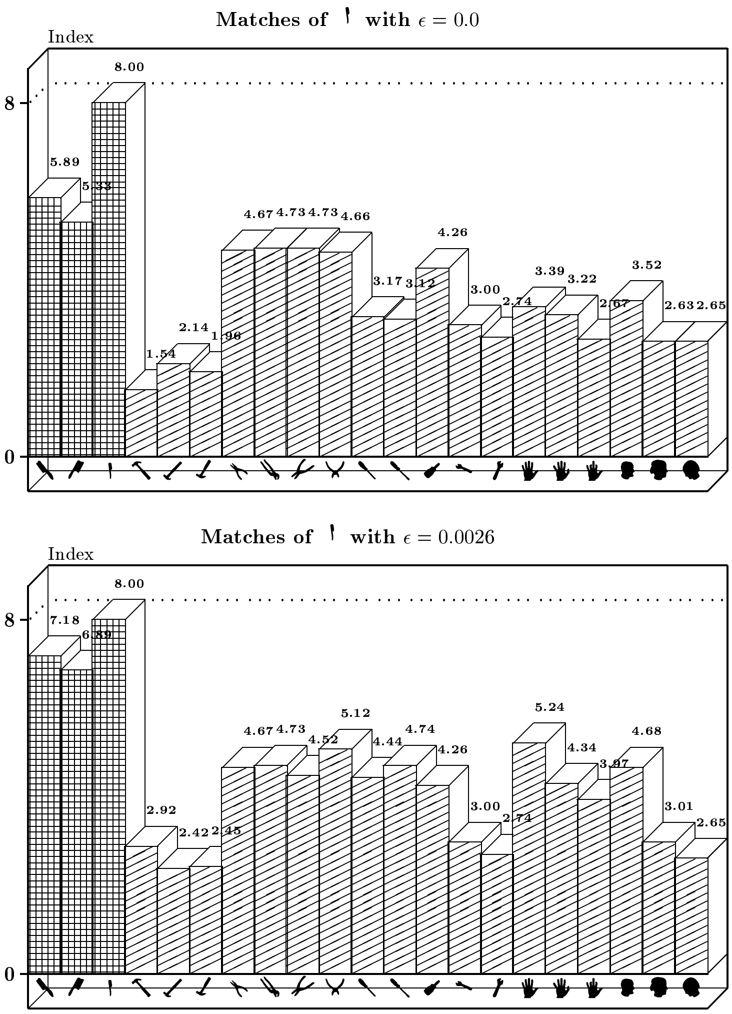


Figure 6: The scores obtained from matching an example prototype object against all other objects in the database (the weighted case), for $\epsilon = 0$ and $\epsilon = 0.0026$. Compare with Figure 5.

pages 298–307, Stanford, CA, 1973.

[2] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J, 1982.

[3] H. G. Barrow and R. M. Burstall. Subgraph isomorphism, matching relational structures, and maximal cliques. *Inform. Process. Lett.*, 4(4):83–84, 1976.

[4] M. Bartoli, *Attributed Tree Homomorphism Using Association Graphs, with Application to Computer Vision* (in Italian). Masters Degree Thesis, Department of Computer Science, University of Venice, Italy, March 1999.

[5] R. C. Bolles and R. A. Cain. Recognizing and locating partially visible objects: The locus-feature-focus method. *Int. J. Robotics Res.*, 1(3):57–82, 1982.

[6] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization (Vol. 4)*, D.-Z. Du and P. M. Pardalos (Eds.), Kluwer Academic Publishers, Boston, MA, 1999 (to appear).

[7] I. M. Bomze. Evolution towards the maximum clique. *J. Global Optim.*, 10:143–164, 1997.

[8] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator equations. *Submitted for publication*, 1999.

[9] M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[10] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision: Vol. 2*. Addison-Wesley, Reading, MA, 1993.

[11] F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

[12] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(11):1168–1180, 1989.

[13] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

[14] T.-L. Liu, D. Geiger, and R. V. Kohn. Representation and self-similarity of shapes. In *Sixth International Conference on Computer Vision*, 1998.

[15] D. Marr and K. H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. R. Soc. Lond. B*, 200:269–294, 1978.

[16] P. M. Pardalos and J. Xue. The maximum clique problem. *J. Global Optim.*, 4:301–328, 1994.

[17] M. Pelillo. Relaxation labeling networks for the maximum clique problem. *J. Artif. Neural Networks*, 2:313–328, 1995.

[18] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8), 1999 (in press).

[19] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Anal. Machine Intell.*, accepted for publication.
A preliminary version appears in: *Computer Vision—ECCV'98*, H. Burkhardt and B. Neumann (Eds.), (Lecture Notes in Computer Science, vol. 1407). Springer-Verlag, Berlin, pp. 3–16, 1998.

[20] F. Pla and J. A. Marchant. Matching feature points in image sequences through a region-based method. *Comp. Vision Image Understanding*, 66:271–285, 1997.

[21] B. Radig. Image sequence analysis using relational structures. *Pattern Recognition*, 17:161–167, 1984.

[22] H. Samet. *Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.

[23] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, to appear, 1999.

[24] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 507–513, 1996.

[25] V. Venkateswar and R. Chellappa. Hierarchical stereo and motion correspondence using feature groupings. *Int. J. Comp. Vision*, 15:245–269, 1995.