Attributed Tree Matching and Maximum Weight Cliques

Marcello Pelillo* Kaleem Siddiqi† Steven W. Zucker‡

Abstract

A classic way of matching relational structures consists of finding a maximum clique in a derived "association graph." However, it is not clear how to apply this approach to problems where the graphs are hierarchically organized, i.e. are trees, since maximum cliques are not constrained to preserve the partial order. We have recently provided a solution to this problem by constructing the association graph using the graph-theoretic concept of connectivity. In this paper, we extend the approach to the problem of matching attributed trees. Specifically, we show how to derive a "weighted" association graph, and prove that the attributed tree matching problem is equivalent to finding a maximum weight clique in it. We then formulate the maximum weight clique problem in terms of a continuous optimization problem, which we solve using "replicator" dynamical systems developed in theoretical biology. This formulation is attractive because it can motivate analog and biological implementations. We illustrate the power of the approach by matching articulated and deformed shapes described by shock trees.

Keywords: Structural matching, attributed trees, weighted cliques, energy minimization methods.

*Dipartimento di Matematica Applicata e Informatica, Università "Ca' Foscari" di Venezia, Via Torino 155, 30173 Venezia Mestre, Italy, E-mail: pelillo@dsi.unive.it

†**(Corresponding Author)** School of Computer Science & Center for Intelligent Machines McGill University, 3480 University Street, Montréal, PQ, Canada H3A 2A7, E-mail: siddiqi@cim.mcgill.ca

‡Department of Computer Science & Center for Computational Vision & Control, Yale University, New Haven, CT 06520-8285, E-mail: zucker-steven@cs.yale.edu

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, 15]). A well known approach to solving this problem consists of transforming it into the 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, 18, 26, 32, 31, 39, 41]. 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 [13], powerful heuristics have been developed which efficiently find good approximate solutions [27].

In many computer vision problems, however, relational structures are organized in a hierarchical manner, i.e., are *trees* (see, for example, [24, 33, 35, 38, 42]). Since in the standard association graph formulation the solutions are not constrained to preserve the required partial order, it is not clear how to apply the framework in these cases. The extension of association graph techniques to tree matching problems is therefore of considerable interest. To illustrate the difficulties with the standard formulation, consider the problem of finding the largest subtree in the left tree of Figure 1 which is isomorphic to a subtree in the right tree. Up to permutations, the correct solution is clearly given by 3 $\rightarrow$ a, 4 $\rightarrow$ b, 5 $\rightarrow$ c, 6 $\rightarrow$ d, 7 $\rightarrow$ f, and 8 $\rightarrow$ g. In other words, the subtree rooted at node 3 is matched against that rooted at node a in the tree on the right. However, using the standard association graph formulation (*cfr.* [2, p. 366]), it is easily verified that the solutions induced by the maximum cliques correspond (up to permutations) to the following: 2 $\rightarrow$ h, 3 $\rightarrow$ a, 4 $\rightarrow$ b, 5 $\rightarrow$ c, 6 $\rightarrow$ d, 7 $\rightarrow$ f, and 8 $\rightarrow$ g, which, while perfectly in accordance with the usual subgraph isomorphism constraints, *does* violate the requirement that the matched subgraphs be trees (note, in fact, that nodes 2 and h are isolated from the rest of the matched subtrees).

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 [30]. 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

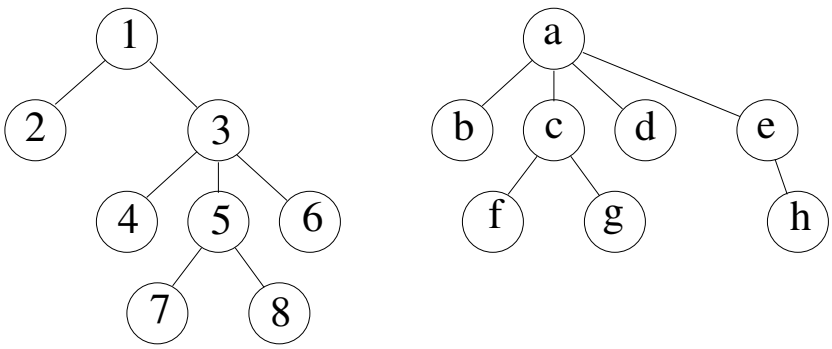

Figure 1: An example of matching two trees. In the standard formulation of the association graph, the maximum cliques do not preserve the hierarchical structure of the two trees.

subtree isomorphisms. As an obvious corollary, the computational complexity of finding a maximum clique in such graphs is therefore the same as the subtree isomorphism problem, which is known to be polynomial in the number of nodes [13].

In this paper, we expand on this framework to the problem of matching attributed trees, i.e., trees whose nodes have one or more associated attributes. Specifically, we cast the attributed tree matching problem as the equivalent problem of finding a maximum weight clique in a "weighted" association graph. We then use a recent generalization of the Motzkin-Straus theorem [25] to formulate the maximum weight clique problem as a quadratic programming problem. To (approximately) solve it we employ *replicator equations*, a class of continuous- and discrete-time dynamical systems developed and studied in various branches of biomathematics [17, 40]. This formulation turns out to be particularly attractive because can motivate simple parallel analog implementations, and also has a certain biological plausibility. We illustrate the power of the approach via several examples of matching articulated and deformed shapes described by *shock* trees [38].

2 Matching Attributed Trees Using Weighted Association Graphs

2.1 Notations and definitions

Before going into the details of the proposed framework, we need to introduce some graph-theoretical notations and definitions. More details can be found in standard textbooks of graph theory, such as [16]. 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 $u_0 u_1 \ldots u_n$ such that for all $i = 1 \ldots n$, $u_{i-1} \sim u_i$; 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). Given a subset of nodes $C \subseteq V$, the *induced subgraph* $G[C]$ is the graph having C as node set, and two nodes are adjacent in $G[C]$ if and only if they are adjacent in G. A *weighted graph* is a triple $G = (V, E, \omega)$, where (V, E) is a graph and ω is a function which assigns a positive weight $\omega(u)$ to each node $u \in V$.

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. An *attributed tree* is a (rooted) tree each node of which has an associated vector of symbolic and/or numeric attributes. Formally, it is a triple $T = (V, E, \alpha)$, where (V, E) is the "underlying" tree and α is a function which assigns an attribute vector $\alpha(u)$ to each node u in the tree.

2.2 Deriving the weighted 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 bijection $\phi : H_1 \to H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is called a *subtree isomorphism* if it preserves the adjacency and hierarchical relations between the nodes and if, in addition, the induced subgraphs $T_1[H_1]$ and $T_2[H_2]$ are trees. The former condition amounts to stating that, given $u, v \in H_1$, we have $u \sim v$ if and only if $\phi(u) \sim \phi(v)$, and u is the parent of v if and only if $\phi(u)$ is the parent of $\phi(v)$.

It is clear that in matching two attributed trees, our objective is to find an isomorphism 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 $\phi : H_1 \to H_2$ is a subtree isomorphism between two attributed trees T_1 and T_2, the overall similarity between the induced subtrees $T_1[H_1]$ and $T_2[H_2]$ can be defined as follows:

$$S(\phi) = \sum_{u \in H_1} \sigma(\alpha_1(u), \alpha_2(\phi(u))) \ . \tag{1}$$

The isomorphism ϕ is called a *maximal similarity subtree isomorphism* if there is no other subtree isomorphism $\phi' : H_1' \to H_2'$ such that H_1 is a strict subset of H_1' and $S(\phi) < S(\phi')$. It is called a *maximum similarity subtree isomorphism* if $S(\phi)$ is largest among all subtree isomorphisms between T_1 and T_2.

We now introduce the notion of a *path-string* which turns out to be of pivotal importance for our subsequent development.

Definition 1 *Let u and v be two distinct nodes of a rooted tree T, and let $u = x_0 x_1 \ldots x_n = v$ be the (unique) path joining them. The* path-string *of u and v, denoted by $\mathrm{str}(u, v)$, is the string $s_1 s_2 \ldots s_n$ on the alphabet $\{-1, +1\}$ where, for all $i = 1 \ldots n$, $s_i = \mathrm{lev}(x_i) - \mathrm{lev}(x_{i-1})$.*

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 you have to do in order to reach v, starting from u. It may be thought of as the degree of relationship between two relatives in a "family" tree.

The (weighted) *association graph* of two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ is the weighted graph $G = (V, E, \omega)$ where

$$V = V_1 \times V_2 \ , \tag{2}$$

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

$$(u, w) \sim (v, z) \Leftrightarrow \mathrm{str}(u, v) = \mathrm{str}(w, z), \tag{3}$$

and, for all $(u, v) \in V$, the weight function is defined as

$$\omega(u, v) = \sigma(\alpha_1(u), \alpha_2(v)) \ . \tag{4}$$

Intuitively, (3) says that two nodes (u, w) and (v, z) are adjacent in G, if and only if the relationship between u and v in T_1 is the same as that between w and z in T_2.

A subset of vertices of G is said to be a *clique* if all its nodes are mutually adjacent. A *maximal weight* clique is one which is not contained in any other clique having larger weight, while a *maximum weight* clique is a clique having largest weight. The maximum weight clique problem is to find a maximum weight clique of G [27]. The following result, which is the basis of the work reported here, establishes a one-to-one correspondence between our attributed tree matching problem and the maximum weight clique problem (see [30] for proof).

Theorem 1 *Any maximal (maximum) similarity subtree isomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding association graph, and vice versa.*

The next proposition provides us with a straightforward criterion to construct the association graph.

Proposition 1 *Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two rooted trees, $u, v \in V_1$, and $w, z \in V_2$. Then,* $\mathrm{str}(u, v) = \mathrm{str}(w, z)$ *if and only if the following two conditions hold:*

(a) $d(u,v) = d(w,z)$

(b) $\mathrm{lev}(u) - \mathrm{lev}(v) = \mathrm{lev}(w) - \mathrm{lev}(z)$

This result allows us to efficiently derive the association graph by using a classical representation for graphs, i.e., the so-called *distance matrix* (see, e.g., [16]) which, for an arbitrary graph $G = (V, E)$ of order n, is the $n \times n$ matrix $D = (d_{ij})$ where $d_{ij} = d(u_i, u_j)$, the distance between nodes u_i and u_j.

3 Tree Matching Replicator Equations

Let $G = (V, E, \omega)$ be an arbitrary weighted graph of order n, and let S_n denote the standard simplex of $\mathbb{R}^n$:

$$S_n = \{ \ \mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{e}'\mathbf{x} = 1 \text{ and } x_i \geq 0, \ i = 1 \ldots n \ \}$$

where $\mathbf{e}$ is the vector whose components equal 1, and a prime denotes transposition. Given a subset of vertices C of G, we will denote by $\mathbf{x}^c$ its *characteristic vector* which is the point in S_n defined as

$$x_i^c = \begin{cases} \omega(u_i)/\Omega(C), & \text{if } u_i \in C \\ 0, & \text{otherwise} \end{cases}$$

where $\Omega(C) = \sum_{u_j \in C} \omega(u_j)$ is the total weight on C.

Now, consider the following class of $n \times n$ symmetric matrices:

$$\mathcal{M}(G) = \left\{ A = (a_{ij})_{1 \le i,j \le n} : a_{ii} = \frac{1}{2\omega(u_i)},\ a_{ij} = 0 \text{ if } u_i \sim u_j,\ a_{ij} = a_{ji} \ge a_{ii} + a_{jj} \text{ otherwise} \right\}$$

and the quadratic function

$$f(\mathbf{x}) = \mathbf{x}' A \mathbf{x} \tag{5}$$

where $A \in \mathcal{M}(G)$. A point $\mathbf{x}^* \in S_n$ is said to be a *global* minimizer of f in S_n if $f(\mathbf{x}^*) \le f(\mathbf{x})$, for all $\mathbf{x} \in S_n$. It is said to be a *local* minimizer if there exists an $\varepsilon > 0$ such that $f(\mathbf{x}^*) \le f(\mathbf{x})$ for all $\mathbf{x} \in S_n$ whose distance from $\mathbf{x}^*$ is less than ε, and if $f(\mathbf{x}^*) = f(\mathbf{x})$ implies $\mathbf{x}^* = \mathbf{x}$, then $\mathbf{x}^*$ is said to be a *strict* local minimizer.

The following theorem expands on a recent result by Gibbons *et al.* [14], which in turn generalizes the so-called Motzkin-Straus theorem [25]. This allows us to formulate the maximum weight clique problem as a quadratic program, thereby switching from the discrete to the continuous domain (see [7, 9] for proof).

Theorem 2 *Let $G = (V, E, \omega)$ be an arbitrary weighted graph and consider a matrix $A \in \mathcal{M}(G)$. Then, the following hold:*

(a) A vector $\mathbf{x} \in S_n$ is a local minimizer of f on S_n if and only if $\mathbf{x} = \mathbf{x}^c$, where C is a maximal weight clique of G.

(b) A vector $\mathbf{x} \in S_n$ is a global minimizer of f on S_n if and only if $\mathbf{x} = \mathbf{x}^c$, where C is a maximum weight clique of G.

Moreover, all local (and hence global) minimizers of f on S_n are strict.

Unlike the Gibbons *at al.*'s formulation, which is plagued by the presence of "spurious" solutions [9], the previous result guarantees that *all* minimizers of f on S_n are strict, and

are characteristic vectors of maximal/maximum cliques in the graph. In a formal sense, therefore, a one-to-one correspondence exists between maximal/maximum weight cliques and local/global minimizers of f on S_n

A "distinguished" member of the class $\mathcal{M}(G)$, which is the one used in our experiments, is as follows:

$$a_{ij} = \begin{cases} \frac{1}{2\omega(u_i)} & \text{if } i = j \ , \\ 0 & \text{if } i \neq j \text{ and } u_i \sim u_j \ , \\ \frac{1}{2\omega(u_i)} + \frac{1}{2\omega(u_j)} & \text{otherwise .} \end{cases} \tag{6}$$

For the unweighted case, where $\omega(u) = 1$ for all $u \in V$, this formulation goes back to a different characterization of maximal cliques which has been thoroughly studied in [6, 8].

We now turn our attention to a class of simple dynamical systems that we use for solving our quadratic optimization problem. Let W be a non-negative real-valued $n \times n$ matrix, and consider the following dynamical system:

$$\dot{x}_i(t) = x_i(t) \left[(W\mathbf{x}(t))_i - \mathbf{x}(t)' W \mathbf{x}(t) \right] , \quad i = 1 \ldots n \tag{7}$$

where a dot signifies derivative w.r.t. time t, and its discrete-time counterpart

$$x_i(t+1) = x_i(t) \frac{(W\mathbf{x}(t))_i}{\mathbf{x}(t)' W \mathbf{x}(t)} , \quad i = 1 \ldots n \ . \tag{8}$$

It is readily seen that the simplex S_n is invariant under these dynamics, which means that every trajectory starting in S_n will remain in S_n for all future times. Moreover, it turns out that their *stationary points*, i.e. the points satisfying $\dot{x}_i(t) = 0$ for (7) or $x_i(t+1) = x_i(t)$ for (8), coincide and are the solutions of the equations

$$x_i[(W\mathbf{x})_i - \mathbf{x}' W \mathbf{x}] = 0 , \quad i = 1 \ldots n \ . \tag{9}$$

A stationary point $\mathbf{x}$ is said to be *asymptotically stable* if every solution to (7) or (8) which starts close enough to $\mathbf{x}$, will converge to $\mathbf{x}$ as $t \to \infty$.

Both (7) and (8) are called *replicator equations* in theoretical biology, since they are used to model evolution over time of relative frequencies of interacting, self-replicating entities [17]. The discrete-time dynamical equations turn out to be a special case of a general class of dynamical systems introduced by Baum and Eagon [4] in the context of Markov chains theory. They also represent an instance of the original heuristic Rosenfeld-Hummel-Zucker relaxation labeling algorithm [34], whose dynamical properties have recently been clarified [29] (specifically, it corresponds to the 1-object, n-label case).

We are now interested in the dynamical properties of replicator dynamics; it is these properties that will allow us to solve our original tree matching problem.

Theorem 3 *If $W = W'$ then the function $\mathbf{x}(t)'W\mathbf{x}(t)$ is strictly increasing with increasing t along any non-stationary trajectory $\mathbf{x}(t)$ under both continuous-time (7) and discrete-time (8) replicator dynamics. Furthermore, any such trajectory converges to a stationary point. Finally, a vector $\mathbf{x} \in S_n$ is asymptotically stable under (7) and (8) if and only if $\mathbf{x}$ is a strict local maximizer of $\mathbf{x}'W\mathbf{x}$ on S_n.*

The previous result is known in mathematical biology as the Fundamental Theorem of Natural Selection [11, 17, 40] and, in its original form, traces back to Fisher [12]. As far as the discrete-time model is concerned, it can be regarded as a straightforward implication of the more general Baum-Eagon theorem [4]. The fact that all trajectories of the replicator dynamics converge to a stationary point has been proven more recently [22, 23].

In light of their dynamical properties, replicator equations naturally suggest themselves as a simple heuristic for solving our attributed tree matching problem. Indeed, note that replicator equations are maximization procedures while ours is a minimization problem. However, it is straightforward to see that the problem of minimizing the quadratic form $\mathbf{x}'A\mathbf{x}$ on the simplex is equivalent to that of maximizing $\mathbf{x}'(\gamma\mathbf{e}\mathbf{e}' - A)\mathbf{x}$, where γ is an arbitrary constant. Let $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$ be two attributed trees, and let $G = (V, E, \omega)$ be the corresponding weighted association graph, as defined in Section 2. By letting

$$W = \gamma\mathbf{e}\mathbf{e}' - A \tag{10}$$

where $A = (a_{ij})$ is any matrix in the class $\mathcal{M}(G)$ and $\gamma = \max a_{ij}$, we know that the replicator dynamical systems (7) and (8), starting from an arbitrary initial state, will iteratively maximize the function $\mathbf{x}'W\mathbf{x}$ (and hence minimize $\mathbf{x}'A\mathbf{x}$) over the simplex and will eventually converge with probability 1 to a strict local optimizer which, by virtue of Theorem 2, will then correspond to the characteristic vector of a maximal weight clique in the association graph. As stated in Theorem 1, this will in turn induce a maximal similarity subtree isomorphism between T_1 and T_2.

Clearly, in theory there is no guarantee that the converged solution will be a *global* optimizer, and therefore that it will induce a *maximum* weight isomorphism between the two

original trees. Previous experimental work done on the (unweighted) maximum clique problem [8, 28], and also the results presented in the next section, however, suggest that the basins of attraction of optimal or near-optimal solutions are quite large, and very frequently the algorithm converges to one of them, despite its inherent inability to escape from local optima.

Since the process cannot leave the boundary of the simplex, it is customary to start out the relaxation process from some interior point, a common choice being the barycenter of the state space. This prevents the search from being initially biased in favor of any particular solution.

4 Examples

We illustrate our framework with numerical examples of shape matching. Because of the subtleties associated with generating random trees of relevance to applications in computer vision and pattern recognition, we use a class of example trees derived from a real system. Our representation for shape is based on an abstraction of the *shocks* (or singularities) of a curve evolution process, acting on a simple closed curve in the plane, into a shock tree. We begin by providing some background on the representation (for details see [20, 38]) and then present experimental results on matching shock trees.

4.1 The Shock Tree

In [19, 20] the following evolution equation was proposed for visual shape analysis:

$$\begin{aligned} \mathcal{C}_t &= (1 + \alpha\kappa)\mathcal{N} \\ \mathcal{C}(p,0) &= \mathcal{C}_0(p). \end{aligned} \tag{11}$$

Here $\mathcal{C}(p,t)$ is the vector of curve coordinates, $\mathcal{N}(p,t)$ is the inward normal, p is the curve parameter, and t is the evolutionary time of the deformation. The constant $\alpha \geq 0$ controls the regularizing effects of curvature κ. When α is large, the equation becomes a geometric heat equation; when $\alpha = 0$, the equation is hyperbolic and *shocks* [21], or entropy-satisfying singularities, can form. In the latter case the locus of points through which the shocks migrate is related to Blum's grassfire transformation [10, 20], although significantly more information is available via a "coloring" of these positions. Four types can arise, according to the local variation of the radius function along the medial axis (Figure 2). Intuitively,

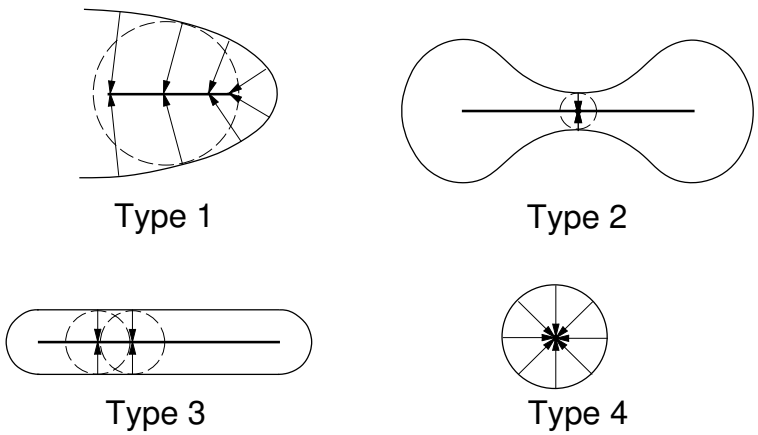

Figure 2: A coloring of shocks into four types [20]. A 1-shock derives from a *protrusion*, and traces out a curve segment of adjacent 1-shocks, along which the radius function varies monotonically. A 2-shock arises at a *neck*, where the radius function attains a strict local minimum, and is immediately followed by two 1-shocks flowing away from it in opposite directions. 3-shocks correspond to an annihilation into a curve segment due to a *bend*, along which the radius function is constant, and a 4-shock an annihilation into a point or a *seed*, where the radius function attains a strict local maximum. The loci of these shocks gives Blum's medial axis.

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 classification of shock positions according to their colors, and an enumeration of the possible local neighborhoods around each shock type is at the heart of the representation.

Shocks of the same type that form a connected component are grouped together to comprise the nodes of a *shock graph*, with the 1-shock groups separated at branch-points of the skeleton. Directed edges in the graph are placed between shock groups that touch one another, such that each parent node contains no shocks that formed prior to any shocks in a child node. This corresponds to a "reversal" in time of the curve evolution process to obtain a hierarchy of connected components. The graph is rooted at a unique vertex #, the children of which are the last shock groups to form, e.g., the palm of the hand shape in Figure 3, with associated nodes 3-004a and 3-004b in the graph. (The letters 'a' and 'b' denote different sides of the same shock group). A key property of the shock graph is that its topological structure is highly constrained because the events that govern the birth, combination and death of shock groups are completely characterized by a shock grammar,

with a small number of rewrite rules [38]. In particular, each shock graph can be reduced to a unique rooted shock tree.

4.2 Experimental Results

We now provide examples of weighted shock tree matching, using the geometric attributes associated with shock tree nodes. 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 [36, 37]. 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

$$\sigma(\alpha(u), \alpha(v)) = 1 - \beta_l \frac{|L(u) - L(v)|}{max(L(u), L(v))} - \beta_r \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{r(i) - r(\xi(i))}{max(r(i), r(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}}$$
$$-\beta_v \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{v(i) - v(\xi(i))}{max(v(i), v(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}} - \beta_{\Delta\theta} \left\{ \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\Delta\theta(i) - \Delta\theta(\xi(i))}{max(\Delta\theta(i), \Delta\theta(\xi(i)))} \right)^2 \right\}^{\frac{1}{2}} \quad (12)$$

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

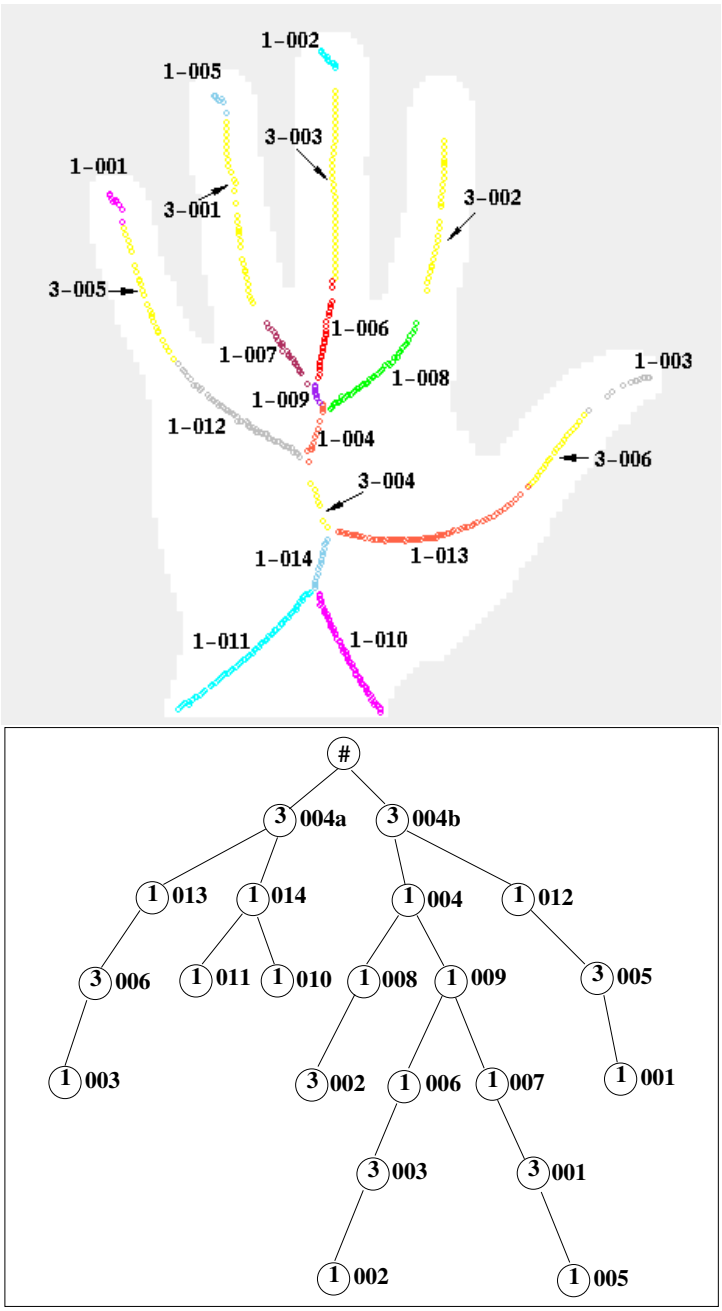

Figure 3: An illustrative example of the shocks obtained from curve evolution (from [38]). TOP: The notation associated with the locus of shock points is of the form shock_type-identifier. BOTTOM: The tree has the shock_type on each node, and the identifier is adjacent. The last shocks to form during the curve evolution process appear under the root node labeled #.

measure is designed to be invariant under rotations and translations of two shapes, and to satisfy the requirements of the weight function discussed in Section 2.

We selected 25 silhouettes representing eight different object classes (Table 1, first column). The tool shapes were taken from the Rutgers Tools database. Each shape was matched against *all* entries in the database, 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 (12), 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 $W \times \frac{1}{2}\{\sum_{i=1}^{n}(m(u_i)/M_1 + m(v_i)/M_2)\}$, where n is the size of the maximal weight clique found, W its weight, 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. The top 8 matches are shown for each query shape, in Table 1. All exemplars in the same class as the query shape are typically within the top 5 matches.

We note that the qualitative results, specifically, the partial ordering of matches in each row of Table 1, compare favorably with those obtained using an alternate approach described in [38]. The latter approach has been applied successfully to the same database of shapes used here, and is a sequential (level by level) approach with back-tracking. At each step an eigenvalue labeling of the adjacency matrix of the subtrees rooted at the two nodes being matched is used. In other words, the cost of matching two nodes incorporates not only their geometries, but a measure of the similarity between their subtrees, which is *global*. Furthermore, the algorithm tolerates noise by allowing for jumps between levels. Hence, strictly speaking, it solves an *approximation* to the subtree isomorphism problem. For these reasons, a one-to-one comparison between the two algorithms is not possible; they solve different problems and the maximal clique formulation invokes much weaker constraints. We are presently investigating the extension of the maximal clique formulation to the case of many-to-many correspondences, which is of particular interest in computer vision and pattern recognition applications where the trees being matched are noisy, and vertices are deleted or added.

Query Shape	Top 8 Attributed Topological Matches							
	1	2	3	4	5	6	7	8
	10.00	6.97	5.89	4.94	4.85	4.83	4.74	4.54
	10.00	6.97	5.33	5.22	5.10	5.04	4.98	4.87
	8.00	5.89	5.33	4.73	4.73	4.67	4.66	4.61
	12.00	8.91	6.82	5.73	4.73	4.26	3.49	3.38
	9.00	6.82	6.78	5.00	4.93	4.68	3.90	3.76
	11.00	8.91	6.78	6.06	4.88	4.34	3.60	3.58
	15.00	10.79	10.26	10.13	8.90	7.83	6.65	6.29
	16.00	9.22	9.21	9.05	8.90	6.68	5.53	5.32
	14.00	10.13	9.61	9.48	9.21	8.20	6.10	6.10
	12.00	8.28	8.20	7.83	7.68	6.68	5.64	5.53
	14.00	10.79	9.94	9.61	9.05	8.28	5.77	5.70
	15.00	10.26	9.94	9.48	9.22	7.68	6.20	6.16
	12.00	8.96	5.15	4.61	4.53	4.52	4.49	4.28
	12.00	8.96	4.99	4.47	4.43	4.36	4.33	3.87
	13.00	6.67	6.20	6.10	5.15	4.99	4.70	4.65
	8.00	5.58	4.68	4.34	4.26	3.16	3.01	2.93
	10.00	6.06	5.73	5.58	5.00	3.90	2.94	2.73
	21.00	19.23	18.49	7.37	6.45	6.22	6.20	6.20
	22.00	18.49	17.83	7.35	6.29	6.21	6.16	6.10
	23.00	19.23	17.83	7.44	6.92	6.67	6.16	6.10
	12.00	7.44	7.37	7.35	5.93	5.63	5.22	4.64
	12.00	9.67	5.93	5.35	5.28	5.28	4.70	3.34
	12.00	9.67	5.63	5.21	5.15	5.11	4.57	3.23
	40.00	12.44	6.92	6.65	6.45	6.21	5.52	5.32
	31.00	12.44	6.10	6.04	6.02	5.38	5.87	5.28

Table 1: A tabulation of the top 8 attributed topological matches for each query. The scores indicate the weight of the maximal clique, multiplied by the average of the total relative mass of nodes in each tree matched (see text).

5 Conclusions

We have developed a formal approach for matching attributed trees by constructing a weighted association graph whose maximum weight cliques are in one-to-one correspondence with maximum similarity subtree isomorphisms. The framework is general and can be applied in a variety of computer vision domains: we have demonstrated its potential for shape matching. The solution is found by using a dynamical system, which makes it amenable to hardware implementation and offers the advantage of biological plausibility.

Acknowledgements We thank Sven Dickinson for the use of shapes from the Rutgers Tools database. This work was supported by Consiglio Nazionale delle Ricerche (Italy), NSF and AFOSR.

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.*, 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] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

[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. Evolution towards the maximum clique. *J. Global Optim.*, 10:143–164, 1997.

[7] I. M. Bomze. On standard quadratic optimization problems. *J. Global. Optim.*, submitted, 1997.

[8] I. M. Bomze, M. Pelillo, and R. Giacomini. Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale. In I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, editors, *Developments in Global Optimization*, pages 95–108, Dordrecht, The Netherlands, 1997. Kluwer.

[9] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique: An evolutionary game theory approach. In preparation, 1998.

[10] R. Brockett and P. Maragos. Evolution equations for continuous-scale morphology. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA, March 1992.

[11] J. F. Crow and M. Kimura. *An Introduction to Population Genetics Theory.* Harper & Row, New York, 1970.

[12] R. A. Fisher. *The Genetical Theory of Natural Selection.* Oxford University Press, London, UK, 1930.

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

[14] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana. Continuous characterizations of the maximum clique problem. *Math. Oper. Res.*, 22:754–768, 1997.

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

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

[17] J. Hofbauer and K. Sigmund. *The Theory of Evolution and Dynamical Systems.* Cambridge University Press, Cambridge, UK, 1988.

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

[19] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Toward a computational theory of shape: An overview. *Lecture Notes in Computer Science*, 427:402–407, 1990.

[20] 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. *Int. J. Comp. Vision*, 15:189–224, 1995.

[21] P. D. Lax. Shock waves and entropy. In E. H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 603–634, New York, 1971. Acad. Press.

[22] V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *J. Math. Biol.*, 17:241–251, 1983.

[23] Y. Lyubich, G. D. Maistrowskii, and Y. G. Ol'khovskii. Selection-induced convergence to equilibrium in a single-locus autosomal population. *Problems of Information Transmission*, 16:66–75, 1980.

[24] 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.

[25] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canad. J. Math.*, 17:533–540, 1965.

[26] H. Ogawa. Labeled point pattern matching by delanauy triangulation and maximal cliques. *Pattern Recognition*, 19:35–40, 1986.

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

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

[29] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *J. Math. Imaging Vision*, 7:309–323, 1997.

[30] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. Technical report, Yale University, Dept. Computer Science, 1997.

[31] 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.

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

[33] H. Rom and G. Medioni. Hierarchical decomposition and axial shape description. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(10):973–981, 1993.

[34] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybern.*, 6:420–433, 1976.

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

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

[37] K. Siddiqi, B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shapes, shocks and wiggles. *Image and Vision Computing*, to appear, 1999.

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

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

[40] J. W. Weibull. *Evolutionary Game Theory.* MIT Press, Cambridge, MA, 1995.

[41] B. Yang, W. E. Snyder, and G. L. Bilbro. Matching oversegmented 3d images using association graphs. *Image Vision Comput.*, 7:135–143, 1989.

[42] S. Zhu and A. L. Yuille. Forms: A flexible object recognition and modelling system. *Int. J. Comp. Vision*, 20(3):187–212, 1996.