

# Matching Hierarchical Structures Using Association Graphs

Marcello Pelillo\*, Kaleem Siddiqi, and Steven W. Zucker

Yale University

Center for Computational Vision & Control

{pelillo-marcello, siddiqi-kaleem, zucker-steven}@cs.yale.edu

**Abstract.** It is well known that the problem of matching two relational structures can be posed as an equivalent problem of finding a maximal clique in a (derived) “association graph.” However, it is not clear how to apply this approach to computer vision problems where the graphs are hierarchically organized, i.e. are trees, since maximal cliques are not constrained to preserve the partial order. Here we provide a solution to the problem of matching two trees, by constructing the association graph using the graph-theoretic concept of connectivity. We prove that in the new formulation there is a one-to-one correspondence between maximal cliques and maximal subtree isomorphisms, and show how to solve the matching problem using simple “replicator” dynamical systems developed in theoretical biology. Such continuous solutions to discrete problems can motivate analog and biological implementations. We illustrate the power of 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. A well-known approach to solve 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, 13, 21, 29, 28, 34, 36]. 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 [10], powerful heuristics have been developed which efficiently find good approximate solutions [22].

In many computer vision problems, however, relational structures are organized in a hierarchical manner, i.e., are *trees* (see, for example, [17, 30, 32, 33,

---

\* Permanent address: Dipartimento di Matematica Applicata e Informatica, Università “Ca’ Foscari” di Venezia, Via Torino 155, 30173 Venezia Mestre, Italy, E-mail: pelillo@dsi.unive.it

37]). 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 (*cf.* [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).

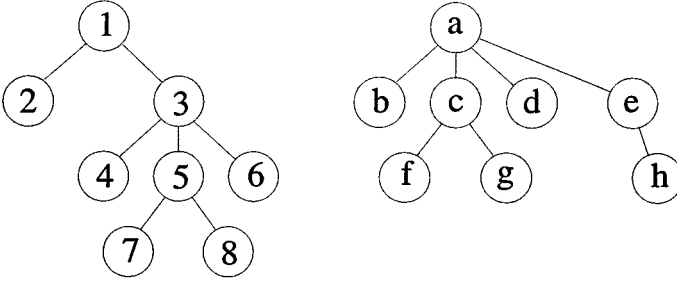
In this paper, we introduce a solution to this problem by providing a novel way of deriving an association graph from two (rooted) trees, based on the graph-theoretic notions of connectivity and the distance matrix. We prove that in the new formulation there is a one-to-one correspondence between maximal (maximum) cliques in the derived association graph and maximal (maximum) 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 [10].

Following the development in [25], we use a recent generalization of the Motzkin-Straus theorem [20] to formulate the maximum clique problem as a quadratic programming problem. To (approximately) solve it we employ *replicator equations*, a class of simple continuous- and discrete-time dynamical systems developed and studied in various branches of biomathematics [12, 35]. We illustrate the power of the approach via several examples of matching articulated and deformed shapes described by *shock* trees [33].

## 2 Tree Isomorphism and Maximal Cliques

### 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 [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



**Fig. 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.

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 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  $\text{lev}(u)$ , 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  $\text{lev}(v) - \text{lev}(u) = +1$ , 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.

## 2.2 Deriving the association graph

Let  $T_1 = (V_1, E_1)$  and  $T_2 = (V_2, E_2)$  be two rooted trees. Any bijection  $\phi : H_1 \rightarrow H_2$ , 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, 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  $\phi(v)$ . A subtree isomorphism is *maximal* if there is no other subtree isomorphism  $\phi' : H'_1 \rightarrow H'_2$  with  $H_1$  a strict subset of  $H'_1$ , and *maximum* if  $H_1$  has largest cardinality. The maximal (maximum) subtree isomorphism problem is to find a maximal (maximum) subtree isomorphism between two rooted trees.

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  $u = x_0x_1 \dots x_n = v$  be the (unique) path joining them. The path-string of  $u$  and  $v$ , denoted by  $\text{str}(u, v)$ , is the string  $s_1s_2 \dots s_n$  on the alphabet  $\{-1, +1\}$  where, for all  $i = 1 \dots n$ ,  $s_i = \text{lev}(x_i) - \text{lev}(x_{i-1})$ .

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 *association graph* of two rooted trees  $T_1 = (V_1, E_1)$  and  $T_2 = (V_2, E_2)$  is the graph  $G = (V, E)$  where

$$V = V_1 \times V_2 \quad (1)$$

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

$$(u, w) \sim (v, z) \Leftrightarrow \text{str}(u, v) = \text{str}(w, z) . \quad (2)$$

Intuitively, 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  $T_2$ . Note that this definition of association graph is stronger than the standard one used for matching arbitrary relational structures [2, 3]. 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 result, which is the basis of the work reported here, establishes a one-to-one correspondence between the maximum subtree isomorphism problem and the maximum clique problem.

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

*Proof (outline).* Let  $\phi : H_1 \rightarrow H_2$  be a maximal subtree isomorphism between rooted trees  $T_1$  and  $T_2$ , and let  $G = (V, E)$  denote the corresponding association graph, as defined above. The maximal clique induced by  $\phi$  is simply the set of vertices  $C_\phi \subseteq V$  defined as:

$$C_\phi = \{(u, \phi(u)) : u \in H_1\} .$$

Intuitively, the fact that  $C_\phi$  is a clique follows from the observation that  $\phi$  maps the path between any two nodes  $u$  and  $v$  onto the path joining  $\phi(u)$  and  $\phi(v)$ . Trivially,  $C_\phi$  is maximal because  $\phi$  is, and this proves the first part of the theorem.

Suppose now that  $C = \{(u_1, w_1), \dots, (u_n, w_n)\}$  is a maximal clique of  $G$ , and define  $H_1 = \{u_1, \dots, u_n\} \subseteq V_1$  and  $H_2 = \{w_1, \dots, w_n\} \subseteq V_2$ . Define  $\phi : H_1 \rightarrow H_2$  as  $\phi(u_i) = w_i$ , for all  $i = 1 \dots n$ . From the definition of the association graph and the hypothesis that  $C$  is a clique, it is simple to see that  $\phi$  is a one-to-one and onto correspondence between  $H_1$  and  $H_2$ , which trivially preserves both the adjacency and the hierarchical relations between nodes. The fact that  $\phi$  is maximal 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. Suppose by contradiction that this is not the case, and let  $u_i, u_j \in H_1$  be two nodes which are not joined by a path in  $T_1[H_1]$ . Since both  $u_i$  and  $u_j$  are nodes of  $T_1$ , however, there must exist a path  $u_i = x_0 x_1 \dots x_m = u_j$  joining them in  $T_1$ . Let  $x^* = x_k$ , for some  $k = 1 \dots m$ , be a node on this path which is not in  $H_1$ . Moreover, let  $y^* = y_k$  be the  $k$ -th node on the path  $w_i = y_0 y_1 \dots y_m = w_j$  which joins  $w_i$  and  $w_j$  in  $T_2$ . It is easy to show that the set  $\{(x^*, y^*)\} \cup C \subseteq V$  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  $\text{str}(u, v)$  can be obtained by concatenating  $\text{str}(u, x)$  and  $\text{str}(x, v)$ .

The “maximum” part of the statement is proved similarly.  $\square$

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,  $\text{str}(u, v) = \text{str}(w, z)$  if and only if the following two conditions hold:*

- (a)  $d(u, v) = d(w, z)$
- (b)  $\text{lev}(u) - \text{lev}(v) = \text{lev}(w) - \text{lev}(z)$

*Proof.* The proposition is a straightforward consequence of the following two observations. Let  $u$  and  $v$  be any two nodes in a tree, and let  $\text{str}(u, v) = s_1 s_2 \dots s_n$  be the corresponding path-string. Then we have: (1)  $\text{lev}(u) - \text{lev}(v) = \sum_i s_i$ , and (2)  $s_i = +1$  implies  $s_j = +1$  for all  $j \geq i$ .  $\square$

This results 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., [11]) 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  $G = (V, E)$  be an arbitrary graph of order  $n$ , and let  $S_n$  denote the standard simplex of  $\mathbb{R}^n$ :

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

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

$$x_i^c = \begin{cases} 1/|C|, & \text{if } i \in C \\ 0, & \text{otherwise} \end{cases}$$

where  $|C|$  denotes the cardinality of  $C$ .

Now, consider the following quadratic function

$$f(\mathbf{x}) = \mathbf{x}'A\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{x} \quad (3)$$

where  $A = (a_{ij})$  is the adjacency matrix of  $G$ , i.e., the  $n \times n$  symmetric matrix defined as

$$a_{ij} = \begin{cases} 1, & \text{if } v_i \sim v_j \\ 0, & \text{otherwise} \end{cases}$$

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

The following theorem, recently proved by Bomze [6], expands on the Motzkin-Straus theorem [20], a remarkable result which establishes a connection between the maximum clique problem and certain standard quadratic programs. This has an intriguing computational significance in that it allows us to shift from the discrete to the continuous domain in an elegant manner.

**Theorem 2.** *Let  $C$  be a subset of vertices of a graph  $G$ , and let  $\mathbf{x}^c$  be its characteristic vector. Then the following statements hold:*

- (a)  *$C$  is a maximum clique of  $G$  if and only if  $\mathbf{x}^c$  is a global maximizer of the function  $f$  in  $S_n$ . In this case,  $|C| = 1/2(1 - f(\mathbf{x}^c))$ .*
- (b)  *$C$  is a maximal clique of  $G$  if and only if  $\mathbf{x}^c$  is a local maximizer of  $f$  in  $S_n$ .*
- (c) *All local (and hence global) maximizers of  $f$  in  $S_n$  are strict.*

Unlike the original Motzkin-Straus formulation, which is plagued by the presence of “spurious” solutions [26], the previous result guarantees us that *all* maximizers 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 cliques and local maximizers of  $f$  in  $S_n$  on the one hand, and maximum cliques and global maximizers on the other hand.

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:

$$\dot{x}_i(t) = x_i(t) [(W\mathbf{x}(t))_i - \mathbf{x}(t)'W\mathbf{x}(t)], \quad i = 1 \dots n \quad (4)$$

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

$$x_i(t+1) = x_i(t) \frac{(W\mathbf{x}(t))_i}{\mathbf{x}(t)'W\mathbf{x}(t)}, \quad i = 1 \dots n. \quad (5)$$

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 (4) or  $x_i(t+1) = x_i(t)$  for (5), coincide and are the solutions of the equations

$$x_i[(W\mathbf{x})_i - \mathbf{x}'W\mathbf{x}] = 0, \quad i = 1 \dots n. \quad (6)$$

A stationary point  $\mathbf{x}$  is said to be *asymptotically stable* if every solution to (4) or (5) which starts close enough to  $\mathbf{x}$ , will converge to  $\mathbf{x}$  as  $t \rightarrow \infty$ .

Both (4) and (5) are called *replicator equations* in theoretical biology, since they are used to model evolution over time of relative frequencies of interacting, self-replicating entities [12]. 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 [31], whose dynamical properties have recently been clarified [24] (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 (4) and discrete-time (5) replicator dynamics. Furthermore, any such trajectory converges to a stationary point. Finally, a vector  $\mathbf{x} \in S_n$  is asymptotically stable under (4) and (5) 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 [8, 12, 35] and, in its original form, traces back to Fisher [9]. 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 [15, 16].

In light of their dynamical properties, replicator equations naturally suggest themselves as a simple heuristic for solving the maximal subtree isomorphism

problem. Let  $T_1 = (V_1, E_1)$  and  $T_2 = (V_2, E_2)$  be two rooted trees, and let  $A$  denote the  $N$ -node adjacency matrix of the corresponding association graph  $G$ , as defined in Section 2. By letting

$$W = A + \frac{1}{2}I_N$$

where  $I_N$  is the  $N \times N$  identity matrix, we know that the replicator dynamical systems (4) and (5), starting from an arbitrary initial state, will iteratively maximize the function  $f$  defined in (3) over  $S_N$  and will eventually converge with probability 1 to a strict local maximizer which, by virtue of Theorem 2, will then correspond to the characteristic vector of a maximal clique in the association graph. As stated in Theorem 1, this will in turn induce a maximal subtree isomorphism between  $T_1$  and  $T_2$ .

Clearly, in theory there is no guarantee that the converged solution will be a *global* maximizer of  $f$ , and therefore that it will induce a *maximum* isomorphism between the two original trees. Previous experimental work done on the maximum clique problem [7, 23], 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.

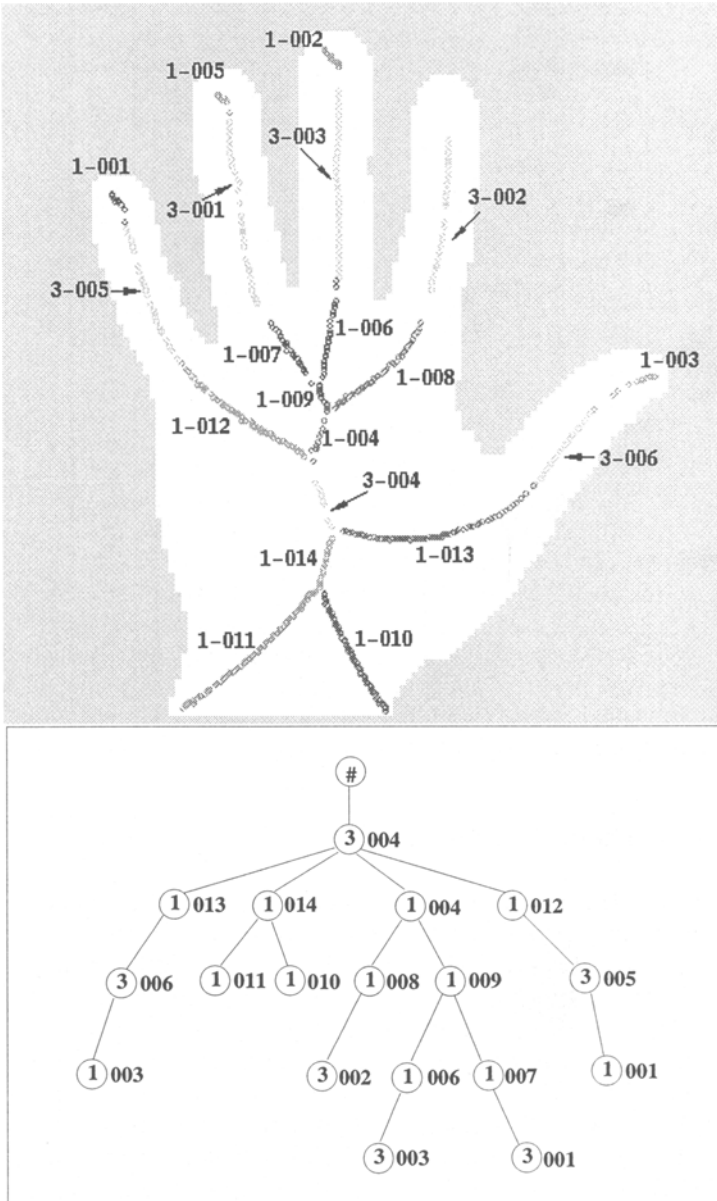
Since the process cannot leave the boundary of  $S_N$ , it is customary to start out the relaxation process from some interior point, a common choice being the barycenter of  $S_N$ , i.e., the vector  $(\frac{1}{N}, \dots, \frac{1}{N})'$ . This prevents the search from being initially biased in favor of any particular solution.

## 4 An Example: Matching Shock Trees

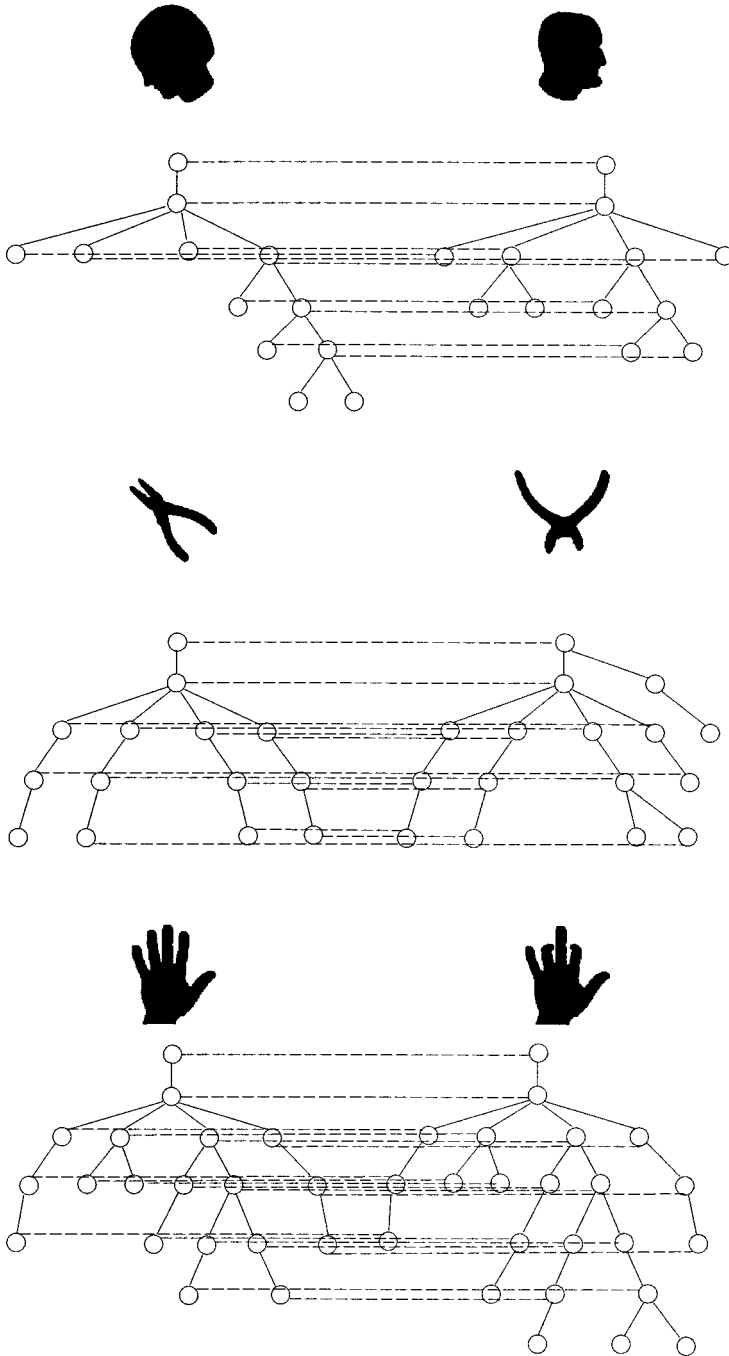
We illustrate our framework with numerical examples of shape matching. We use a *shock graph* representation based on a coloring of the shocks (singularities) of a curve evolution process acting on simple closed curves in the plane [14]. Shocks are grouped into distinct types according to the local variation 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 subgraph isomorphism; see [33] for details. An illustrative example appears in Figure 2. Each graph can be reduced to a unique rooted tree, providing the requisite hierarchical structure for our matching algorithm. The ability of shock trees to discriminate between classes of shapes, using *both* their topologies as well as metric/label information has been examined in [33]. Here we address the unlabeled version of the problem, and examine matching based on topology *alone*. We stress that our goal is to illustrate the power of the hierarchical structure matching algorithm.

We selected 22 silhouettes representing eight different object classes (Table 1, first column); the tools shapes were taken from the Rutgers Tools database. Each





**Fig. 2.** An illustrative example of the shocks obtained from curve evolution (from [33]). 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 shock to form during the curve evolution process is the most “significant,” and this appears under the root node labeled #.



**Fig. 3.** Maximal subtree isomorphisms found for three illustrative examples.

entry was then matched against *all* entries in the database, and the size of the maximal clique found, normalized by the average number of nodes in the two trees, was recorded. Figure 3 shows the maximal subtree isomorphisms (each in one-to-one correspondence with a maximal clique) for three examples. The top 5 matches for each query shape, along with the associated scores, are shown in Table 1. The matching algorithm generally takes only two to three seconds to converge on a Sparc 10.

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 5 matches, and typically in the top 3. It is evident that such a structural matching process for indexing into a database of shapes has great potential; with the addition of geometric information performance can only improve.





































































































































## 5 Conclusions

We have developed a formal approach for matching hierarchical structures by constructing an association graph whose maximal cliques are in one-to-one correspondence with maximal 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. In particular, the relaxation labeling equations are related to putative neuronal implementations [18, 19]. In [27] we extend the present framework to the problem of matching hierarchical structures with attributes. The attributes result in weights being placed on the nodes of the association graph, and a conversion of the maximum clique problem to a maximum weight clique problem.

*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.

Query Shape	Top 5 Topological Matches				
	1	2	3	4	5
	 1.00	 .91	 .81	 .80	 .79
	 1.00	 .81	 .78	 .72	 .70
	 1.00	 .91	 .78	 .72	 .70
	 1.00	 .70	 .57	 .57	 .50
	 1.00	 .70	 .69	 .57	 .50
	 1.00	 1.00	 .83	 .80	 .80
	 1.00	 1.00	 .83	 .80	 .80
	 1.00	 .91	 .86	 .83	 .83
	 1.00	 .96	 .87	 .83	 .81
	 1.00	 .91	 .87	 .84	 .82
	 1.00	 .96	 .91	 .87	 .85
	 1.00	 .92	 .87	 .85	 .81
	 1.00	 .92	 .86	 .82	 .82
	 1.00	 .87	 .86	 .80	 .79
	 1.00	 1.00	 .80	 .76	 .75
	 1.00	 1.00	 .80	 .76	 .75
	 1.00	 .80	 .80	 .80	 .80
	 1.00	 .90	 .87	 .86	 .86
	 1.00	 .95	 .95	 .90	 .86
	 1.00	 .89	 .89	 .80	 .80
	 1.00	 1.00	 .95	 .91	 .89
	 1.00	 1.00	 .95	 .91	 .89

**Table 1.** A tabulation of the top 5 matches for each query. The scores indicate the size of the maximal clique found, normalized by the average number of nodes in the two trees. Note that *only* the topology of the shock trees was used; the addition of geometric information permits finer comparisons [27].

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, 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.
8. J. F. Crow and M. Kimura. *An Introduction to Population Genetics Theory*. Harper & Row, New York, 1970.
9. R. A. Fisher. *The Genetical Theory of Natural Selection*. Oxford University Press, London, UK, 1930.
10. M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
11. F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.
12. J. Hofbauer and K. Sigmund. *The Theory of Evolution and Dynamical Systems*. Cambridge University Press, Cambridge, UK, 1988.
13. R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(11):1168–1180, 1989.
14. 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.
15. V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *J. Math. Biol.*, 17:241–251, 1983.
16. Y. Lyubich, G. D. Maistrovskii, and Y. G. Ol'khovskii. Selection-induced convergence to equilibrium in a single-locus autosomal population. *Problems of Information Transmission*, 16:66–75, 1980.
17. 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.
18. D. Miller and S. W. Zucker. Efficient simplex-like methods for equilibria of non-symmetric analog networks. *Neural Computation*, 4(2):167–190, 1992.
19. D. Miller and S. W. Zucker. Computing with self-excitatory cliques: A model and an application to hyperacuity-scale computation in visual cortex. *Neural Computation*, 1998. To be published.
20. 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.
21. H. Ogawa. Labeled point pattern matching by delaunay triangulation and maximal cliques. *Pattern Recognition*, 19:35–40, 1986.
22. P. M. Pardalos and J. Xue. The maximum clique problem. *J. Global Optim.*, 4:301–328, 1994.
23. M. Pelillo. Relaxation labeling networks for the maximum clique problem. *J. Artif. Neural Networks*, 2:313–328, 1995.
24. M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *J. Math. Imaging Vision*, 7:309–323, 1997.
25. M. Pelillo. A unifying framework for relational structure matching. Submitted, 1997.
26. M. Pelillo and A. Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Networks*, 2:411–420, 1995.
27. M. Pelillo, K. Siddiqi, and S. W. Zucker. Attributed tree matching and maximum weight cliques. Submitted, 1997.

28. 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.
29. B. Radig. Image sequence analysis using relational structures. *Pattern Recognition*, 17:161–167, 1984.
30. H. Rom and G. Medioni. Hierarchical decomposition and axial shape description. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(10):973–981, 1993.
31. A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybern.*, 6:420–433, 1976.
32. H. Samet. *Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.
33. K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. In *Proc. Int. Conf. Comp. Vision*, pages 222–229, Bombay, India, 1998.
34. V. Venkateswar and R. Chellappa. Hierarchical stereo and motion correspondence using feature groupings. *Int. J. Comp. Vision*, 15:245–269, 1995.
35. J. W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, MA, 1995.
36. B. Yang, W. E. Snyder, and G. L. Bilbro. Matching oversegmented 3D images using association graphs. *Image Vision Comput.*, 7:135–143, 1989.
37. S. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modelling system. *Int. J. Comp. Vision*, 20(3):187–212, 1996.