

Computing Approximate Tree Edit-Distance using Relaxation Labeling

Andrea Torsello and Edwin R. Hancock
Dept. of Computer Science, University of York
Heslington, York YO10 5DD, UK

Abstract

This paper investigates an approach to tree edit distance problem with uniform edit cost. We show that any tree obtained with a sequence of cut operations is a subtree of the transitive closure of the original tree. Furthermore, we show that the necessary condition for any subtree to be a solution can be reduced to a clique problem in a derived structure. Using this idea we transform the tree edit distance problem into a series of maximum weight clique problems and then we use relaxation labeling to find an approximate solution.

1. Introduction

Inexact or error-tolerant graph-matching is a problem of pivotal importance in high-level vision. The problem has been studied for almost three decades. Early work on the topic included Barrow and Burstall's idea [1] of locating matches by searching for maximum common subgraphs using the association graph, and the extension of the concept of string edit distance to graph-matching by Fu and his co-workers [6]. These early efforts can be regarded as being goal-directed. More recently, the activity in the area has focused on the use of more principled theoretical ideas. Examples include the use of generative or probabilistic models [5, 18], the use of powerful continuous optimization methods borrowed from the literature on neural networks [8], and the use of linear optimization methods.

Making use of linear optimization methods we find Wang, Zhang and Chirn [17] that approximate the graph edit distance problem with a maximum flow problem, and Barrow and Burstall [1] that transform the maximum common subgraph problem (MCSP) into a max clique problem. The transition from the MCSP to the max clique problem, or maximum complete subgraph problem, is done using a derived structure: the association graph.

Transforming a graph matching problem into a max clique problem opens up to a wide spectrum of new possibilities. A diverse array of very powerful heuristics and theoretical results are available for solving the max clique problem. A

particularly important result is the Motzkin-Straus theorem [9] which allows us to transform the max clique problem into a continuous quadratic programming problem.

In an important series of papers, Bunke has recently shown the intimate relationship between the size of the maximum common subgraph and edit distance [4], in particular, he showed that MCS and graph edit distance computation are equivalent.

While these ideas have been extensively studied for graphs, in this paper we are interested in trees. While trees are a special case of graphs, the connectivity and partial order constraints that they represent require adaptation to be made to generic graph matching techniques so that they may be applied to trees. Furthermore, specific characteristics of trees suggest that posing the tree-matching problem as a variant on graph-matching is not the best approach. In particular, both tree isomorphism and subtree isomorphism problems have efficient polynomial time solutions. Moreover, Tai [15] has proposed a generalization of the string edit distance problem from the linear structure of a string to the non-linear structure of a tree. The resulting tree edit distance differs from the general graph edit distance in that edit operations are carried out only on nodes and never directly on edges. Zhang and Shasha [20] have investigated a special case which involves adding the constraint that the solution must maintain the order of the children of a node. With this order among siblings, they showed that the tree-matching problem is still in P and gave an algorithm to solve it. In subsequent work they showed that the unordered case was indeed an NP hard problem [21]. The problem, though, returns to P when we add the constraint of strict hierarchy, that is when separate subtrees are constrained to be mapped to separate subtrees [19].

For the general case we have to resort to non-linear search algorithms similar to the equivalent for the generic graph matching problem. For instance, Pelillo et al. [11] transform the tree isomorphism problem into a single max clique problem, a technique already used for the generic graph isomorphism problem. They use relaxation labeling to obtain a maximal solution to the max clique problem, and, with it, a maximal tree match.

We draw a number of observations from this review of the relevant literature. First, we see that the computation of the unordered tree edit distance still presents a computational bottleneck. Most of the work reported in the literature investigates the simpler problems of subtree isomorphism or ordered tree edit distance. These problems are addressed in both exact and approximate settings. The goal of our work is therefore to introduce a framework in which we can efficiently approximate the computation of unordered tree edit distance. The

approach is follows. We commence by providing a divide and conquer method for the maximum common subtree by search for maximal cliques of the directed association graph. With this representation to hand, we follow Bomze et al. [2] and use a variant of the Motzkin Straus theorem to convert the maximum weighted clique problem into a quadratic programming problem which can be solved by relaxation labeling. The new tree-matching method is evaluated on the problem of shock-tree matching.

2. Association graph

The phase space we use to represent the matching of nodes is the directed association graph, a variant of the association graph. The association graph is a common structure used in graph matching problems. The nodes of this graph are the Cartesian products of nodes of the graphs to be matched, so that each node represent a possible association, or match, of a node in one graph to a node in the other. The edges of the association graph represent the pairwise constraints of the problem: they represent both connectivity on the original graphs and the feasibility of a solution with the linked associations.

Hierarchical graphs have an order relation induced by paths: given two nodes a and b , (a, b) is in this relation if and only if there is a path from a to b . When the directed graph is acyclical, this relation can be shown to be an (irreflexive) order relation. The use of directed arcs in the association graph allows us to make use of this order. We connect nodes with directed arcs in a way that preserves the ordering of the associated graph. The graph obtained can be shown to be ordered still. Specifically, an association graph for the tree isomorphism problem can be shown to be a forest.

For the exact isomorphism problem (maximum common subgraph) the edges of the association graphs are:

$$(v, v') \rightarrow (u, u') \text{ iff } v \rightarrow u \text{ and } v' \rightarrow u' \quad (1)$$

where v and u are nodes on one graph and v' and u' are nodes on the other.

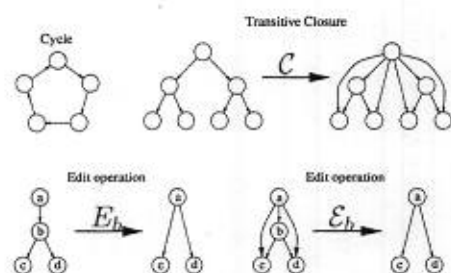


Fig. 1. Some terminology on directed graphs

Proposition 1. *The directed association graph of two directed acyclic graphs (DAGs) G and G' is acyclic.*

Proof. Let us assume that $(u_1, v_1) \rightarrow \dots \rightarrow (u_n, v_n)$ is a cycle. Then, since an arc $(v, v') \rightarrow (u, u')$ in the association graph exists only if the arcs $v \rightarrow u$ and $v' \rightarrow u'$ exist in G and G' respectively, we have that $u_1 \rightarrow \dots \rightarrow u_n$ is a cycle in G and $v_1 \rightarrow \dots \rightarrow v_n$ is a cycle in G' against the hypothesis that they are DAGs. QED

Proposition 2. *The directed association graph of two trees t and t' is a forest.*

Proof. We already know that the association graph is a DAG, we have to show that for each node (u, u') there is at most one node (v, v') such that $(v, v') \rightarrow (u, u')$. Due to the way the association graph is constructed this means that either u or u' must have at most one incoming edge. But t and t' are trees, so both u and u' have at most one incoming edge, namely the one from the parent. QED

The directed association graph can be used to reduce a tree matching problem into subproblems: the best match given the association of nodes v and v' can be found examining only descendents of v and v' . This gives us a divide and conquer solution to the maximum common subtree problem: use the association graph to divide the problem and transform it into maximum bipartite match subproblems, the subproblems can then efficiently be conquered with known polynomial time algorithms. We then extend the approach to the minimum unlabeled tree edit problem and present an evolutionary method to conquer the subproblems. Finally, we present a method to convert the divide and conquer approach into a multi-population evolutionary approach.

3. Inexact tree matching

We want to extend the algorithm to provide us with an error-tolerant tree isomorphism. There is a strong connection between the computation of maximum common subtree and the tree edit distance. In [4] Bunke showed that, under certain constraints applied to the edit-cost function, the maximum common subgraph problem and the graph edit distance problem are computationally equivalent. This is not directly true for trees, because of the added constraint that a tree must be connected. But, extending the concept to the common edited subtree, we can use common substructures to find the minimum cost edited tree isomorphism.

The constraint to the edit-cost function proposed by Bunke in [4] is that the cost of deleting and reinserting the same element with a different label is not greater than the cost of relabeling it. In this way we can find an optimal edit sequence without the need for a relabel operation. We will assume a similar constraint applies to tree edit cost.

With this constraint we are left with only node removal and node insertion operations to be performed on the data tree. Since a node insertion on the data tree is dual to a node removal on the model tree, we can further reduce the number of operations to be performed to only node removal, as long as we perform the operations on both trees.

3.1. Editing the transitive closure of a tree

For each node v of t , we can define an edit operation E_v on the tree and an edit operation \mathcal{E}_v on the closure $\mathcal{C}t$ of the tree t (see Figure 1). In both cases the edit operation removes the node v , all the incoming edges, and all the outgoing edges.

We show that the transitive closure operation and the node removal operation commute, that is we have:

Lemma 3. $\mathcal{E}_v(\mathcal{C}(t)) = \mathcal{C}(E_v(t))$

Proof. If a node is in $\mathcal{E}_v(\mathcal{C}(t))$ it is clearly also in $\mathcal{C}(E_v(t))$. What is left is to show is that an edge (a, b) is in $\mathcal{E}_v(\mathcal{C}(t))$ if and only if it is in $\mathcal{C}(E_v(t))$.

If (a, b) is in $\mathcal{C}(E_v(t))$ then neither a nor b is v and there is a path from a to b in $E_v(t)$. Since the edit operation E_v preserves connectedness and the hierarchy, there must be a path from a to b in t as well. This implies that (a, b) is in $\mathcal{C}(t)$. Since neither a nor b is v , the operation \mathcal{E}_v will not delete (a, b) . Thus (a, b) is in $\mathcal{E}_v(\mathcal{C}(t))$.

If (a, b) is in $\mathcal{E}_v(\mathcal{C}(t))$, then it is also in $\mathcal{C}(t)$, because $\mathcal{E}_v(\mathcal{C}(t))$ is obtained from $\mathcal{C}(t)$ by simply removing a node and some edges. This implies that there is a path from a to b in t and, as long as neither a nor b are v , there is a path from a to b in $E_v(t)$ as well. Thus (a, b) is in $\mathcal{C}(E_v(t))$. Since (a, b) is in $\mathcal{E}_v(\mathcal{C}(t))$, both a and b must be nodes in $\mathcal{E}_v(\mathcal{C}(t))$ and, thus, neither can be v . QED

We call a subtree s of $\mathcal{C}t$ *consistent* if for each node v of s if there cannot be two children a and b so that (a, b) is in $\mathcal{C}t$. In other words, given two nodes a and b , siblings in s , s is consistent if and only if there is no path from a to b in t .

We can, now, prove the following:

Theorem 4. *A tree \hat{t} can be obtained from a tree t with an edit sequence composed of only node removal operations if and only if \hat{t} is a consistent subtree of the DAG Ct .*

Proof. Let us assume that there is an edit sequence $\{E_{v_i}\}$ that transforms t into \hat{t} , then, by virtue of the above lemma, the dual edit sequence $\{\mathcal{E}_{v_i}\}$ transforms Ct into $C\hat{t}$. By construction we have that \hat{t} is a subtree of $C\hat{t}$ and $C\hat{t}$ is a subgraph of Ct , thus \hat{t} is a subtree of Ct . Furthermore, since the node removal operations respect the hierarchy, \hat{t} is a consistent subtree of Ct .

To prove the converse, assume that \hat{t} is a consistent subtree of Ct . If (a, b) is an edge of \hat{t} , then it is an edge on Ct as well, i.e. there is a path from a to b in t and we can define a sequence of edit operations $\{E_{v_i}\}$ that removes any node between a and b in such a path. Showing that the nodes $\{v_i\}$ deleted by the edit sequence cannot be in \hat{t} we show that all the edit operations defined this way are orthogonal. As a result they can be combined to form a single edit sequence that solves the problem.

Let v in \hat{t} be a node in the edited path and let p be the minimum common ancestor of v and a in \hat{t} . Furthermore, let w be the only child of p in \hat{t} that is an ancestor of v in \hat{t} and let q be the only child of p in \hat{t} that is an ancestor of a in \hat{t} . Since a is an ancestor of v in t , an ancestor of v can be a descendant of a , an ancestor of a , or a itself. This means that w has to be in the edited path. Were it not so, then w had to be a or an ancestor of a against the hypothesis that p is the minimum common ancestor of v and a . Since q is an ancestor of a in t and a is an ancestor of w in t , q is an ancestor of w in t , but q and w are siblings in \hat{t} against the hypothesis that \hat{t} is consistent. QED

Using this result, we can show that the minimum cost edited tree isomorphism between two trees t and t' is a maximum common consistent subtree of the two DAGs Ct and Ct' , provided that the node removal cost is uniform. The result can be extended to non-uniform cost, but in this paper we will restrict our scope to the uniform cost case.

The minimum cost edited tree isomorphism is a tree that can be obtained from both model tree t and data tree t' with node removal operations. By virtue of the theorem above, this tree is a consistent subtree of both Ct and Ct' . The tree must be obtained with minimum combined edit cost, and, since the node removal cost is uniform, this implies the minimum number of nodes removed. This implies that the common consistent subtree must retain most nodes, i.e. it must be the maximum common consistent subtree of the two DAGs.

3.2. Cliques and common consistent subtrees

In this section we show that the directed association graph induces a divide and conquer approach to edited tree matching as well. Given two trees t and t' to be matched, we create the directed association graph of the transitive closures Ct and Ct' and we look for a consistent matching tree in the graph. That is we seek a tree in the graph that corresponds to two consistent trees in the transitive closures Ct and Ct' . The maximum such tree corresponds to the maximum common consistent subtree of Ct and Ct' .

In analogy to what we did for the exact matching case, we divide the problem into a maximum common consistent subtree rooted at (v, w) , for each node (v, w) of the association graph. We show that, given the cardinality of the maximum common consistent subtree rooted at each child of (v, w) in the association graph, then we can transform the rooted maximum common consistent subtree problem into a max weighted clique problem. Solving this problem for each node in the association graph and looking for the maximum cardinality rooted common consistent subtree, we can find the solution to the minimum cost edited tree isomorphism problem.

Let us assume that we know the cardinality of the isomorphism for every child of (v, w) in the association graph. We want to find the consistent set of siblings with greatest total cardinality. Let us construct an undirected graph whose nodes consist of the children of (v, w) in the association graph. We connect two nodes (p, q) and (r, s) if and only if there is no path connecting p and r in t and there is no path connecting q and s in t' . This means that we connect two matches (p, q) and (r, s) if and only if they match nodes that are consistent siblings in each tree. Furthermore, we assign to each association node (a, b) a weight equal to the cardinality of the maximum common consistent subtree rooted at (a, b) . The maximum weight clique of this graph will be the set of consistent siblings with maximum total cardinality. The cardinality of the maximum common consistent subtree rooted at (v, w) will be this weight plus one. Furthermore, the nodes of the clique will be the children of (v, w) in the maximum common consistent subtree.

3.3. Heuristics for the maximum weighted clique

In 1965, Motzkin and Strauss [9] showed that the (unweighted) maximum clique problem can be reduced to a quadratic programming problem on the n -dimensional simplex $\Delta = \{\mathbf{x} \in \mathbb{R}^n \mid x_i \geq 0 \text{ for all } i = 1 \dots n, \sum_i x_i = 1\}$, here x_i are the components of vector \mathbf{x} . With this reduction, maximal cliques could be put in correspondence with local maxima of a quadratic function.

Bomze, Pelillo and Stix [2] introduce a regularization factor to the quadratic programming method that generates an equivalent problem with isolated solutions. This quadratic problem is: minimize $f(\mathbf{x}) = \mathbf{x}^T C \mathbf{x}$, subject to $\mathbf{x} \in \Delta$. Where the matrix $C = (c_{ij})_{i,j \in V}$ is defined as

$$c_{ij} = \begin{cases} \frac{1}{2w_i} & \text{if } i = j \\ k_{ij} \geq c_{ii} + c_{jj} & \text{if } (i, j) \notin E, i \neq j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Let us consider a weighted graph $G = (V, E, w)$, where V is the set of nodes, E the set of edges, and $w : V \rightarrow \mathbb{R}$ a weight function that assigns a weight to each node. Given a set $S \subseteq V$ and its characteristic vector \mathbf{x}^S defined as

$$x_i^S = \begin{cases} \frac{w(i)}{\sum_{j \in S} w(j)} & \text{if } i \in S, \\ 0 & \text{otherwise,} \end{cases}$$

S is a maximum (maximal) weight clique if and only if \mathbf{x}^S is a global (local) minimizer for the quadratic problem. Furthermore, if \mathbf{x} is a minimum then it is the characteristic vector for a set of nodes.

To solve the quadratic problem we transform it into the equivalent maximization problem: maximize $\mathbf{x}^T (\gamma \mathbf{e} \mathbf{e}^T - C) \mathbf{x}$, subject to $\mathbf{x} \in \Delta$. Where $\mathbf{e} = (1, \dots, 1)^T$ is the vector with every component equal to 1 and γ is a positive scaling constant. Given the equivalent maximization formulation, we follow [2] and use relaxation labeling as a local maximizer for the problem.

Relaxation labeling is a evidence combining process developed in the framework of constraint satisfaction problems. Its goal is to find a classification p that satisfies pairwise constraints and interactions between its elements. The process is determined by the update rule

$$p_i^{t+1}(\lambda) = \frac{p_i^t(\lambda) q_i^t(\lambda)}{\sum_{\mu} p_i^t(\mu) q_i^t(\mu)}, \quad (3)$$

where the compatibility component is $q_i(\lambda) = \sum_{j=1}^n \sum_{\mu=1}^m \tau_{ij}(\lambda, \mu) p_j(\mu)$.

In [10] Pelillo showed that the function $A(\mathbf{p}) = \sum_{i,\lambda} p_i(\lambda) q_i(\lambda)$ is a Lyapunov function for the process, i.e. $A(\mathbf{p}^{t+1}) \geq A(\mathbf{p}^t)$, with equality if and only if \mathbf{p}^t is stationary.

3.4. Putting it all together

In the previous sections we proved that the maximum edited tree isomorphism problem can be reduced to nm maximum weight clique problem and we have

given an iterative process that is guaranteed to find maximal weight cliques. In this section we will show how to use these ideas to develop a practical algorithm. A direct way is to use the relaxation labeling dynamics starting from the leaves of the directed association graph and propagate the result upwards in the graph using the weight of the extracted clique to initialize the compatibility matrix of every parent association. For each subproblem the compatibility coefficients are initialized as $R_{(v,w)} = \gamma \mathbf{e} \mathbf{e}^T - C$.

This approach imposes a sequentiality to an otherwise highly parallel algorithm. An alternative can be obtained transforming the problem into a single multi-object labeling process. With this approach we set up a labeling problem with one object per node in the association graph, and at each iteration we update the label distribution for each object. We, then, update the compatibility matrices according to the new weight estimate.

This multi-object approach uses the fact that the compatibility matrix for one rooted matching subproblem does not depend upon which nodes are matched below the root, but only on the number of matches. That is, to solve one subproblem we need to know only the weight of the cliques rooted at the children, not the nodes that form the clique. Using Gibbons' approach [7], we get the weight of the clique with the formula $\frac{1}{\mathbf{x}^T B \mathbf{x}}$, where \mathbf{x} is the characteristic vector of the clique and the matrix B is defined as: $\{b_{ii} = \frac{1}{w_i}, b_{ij} = \frac{b_{ii} + b_{jj}}{2}$ if $(i, j) \in E, b_{ij} = 0$ otherwise $\}$. This allows us to generate an estimate of the clique at each iteration: given the current distribution of label probability \mathbf{p} for the subproblem rooted at (v, w) , we estimate the number of nodes matched under (v, w) as $\frac{1}{\mathbf{p}^T B \mathbf{p}}$, and thus we assign to (v, w) the weight $\frac{1}{\mathbf{p}^T B \mathbf{p}} + 1$.

Another possible variation to the algorithm can be obtained using different initial assignments for the label distribution of each subproblem.

A common approach is to initialize the assignment with a uniform distribution so that we have an initial assignment close to the baricenter of the simplex. A problem with this approach is that the dimension of the basin of attraction of one maximal clique grows with the number of nodes in the clique.

With our problem decomposition the wider cliques are the ones that map nodes at lower levels. As a result the solution will be biased towards matches that are very low on the graph, even if these matches require cutting a lot of nodes and are, thus, less likely to give an optimum solution.

A way around this problem is to choose an initialization that assigns a higher initial likelihood to matches that are higher up on the subtree. In our experiments we decided to initialize the probability of the association (a, b) for the subproblem rooted at (u, v) as $p_{(u,v)}(a, b) = e^{-(d_a + d_b + \epsilon)}$, where d_a is the depth of a with respect to u , d_b is the depth of b with respect to v , and ϵ is a small

perturbation. Of course, we then renormalize $p_{(u,v)}$ to ensure that it is still in the simplex.

4. Experimental results

We evaluate the new tree-matching method on the problem of shock-tree matching. characterize the shape of the original boundary. Here we follow Zucker, Siddiqi, and others, by labeling points on the skeleton using so-called shock-labels [14]. According to this taxonomy of local differential structure, there are different classes associated with behavior of the radius of the osculating circle from the skeleton to the nearest pair of boundary points. The so-called shocks distinguish between the cases where the local osculating circle has maximum radius, minimum radius, constant radius or a radius which is strictly increasing or decreasing. We abstract the skeletons as trees in which the level in the tree is determined by their time of formation [12, 14]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy. While this temporal notion of relevance can work well with isolated shocks (maxima and minima of the radius function), it fails on monotonically increasing or decreasing shock groups. To give an example, a protrusion that ends on a vertex will always have the earliest time of creation, regardless of its relative relevance to the shape.

For our experiments we used a database consisting of 13 shapes. For each shape in the database, we computed the maximum edited isomorphism with the other shapes and compared the match using a "goodness" measure. The measure was defined as the average fraction of nodes matched, that is, $W(t_1, t_2) = \frac{1}{2} \left(\frac{\#i}{\#t_1} + \frac{\#i}{\#t_2} \right)$, where $\#t$ indicates the number of nodes in the tree t .

We conducted the experiment with 4 different variants of the algorithm: the multi step, single object update rule with uniform and exponential initial probability assignment, and the single step, multiple object update rule with uniform and exponential initial probability assignment.

In figure 2 we tabulate the top 5 matches in order of goodness for the multi step, single object update rule with exponential initial assignment. The numbers are, from top to bottom, multi-step with exponential initial probability assignment, single-step with exponential initial probability assignment, multi-step with uniform assignment, and single-step with uniform assignment.

As we can see the best match for every shape is the shape itself, and similar shapes have always a high position, usually second.

Our edit distance approach compares favorably against the similar shock graph experiments in [11]. In particular our approach seems to capture better the

perceptual similarity between the shapes of horses and hands. In fact, initial results with multi-dimensional scaling show that the different shape classes form distinct pairwise clusters, and similar shape classes are in close proximity to one-another.

5. Conclusions

In this paper we have investigated a purely structural approach to tree matching. We based the work on to tree edit distance framework constraining it to uniform edit cost. We show that any tree obtained with a sequence of cut operation is a subtree of the transitive closure of the original tree. Furthermore we show that the necessary condition for any subtree to be a solution can be reduced a clique problem in a derived structure. Using this idea we transform the tree edit distance problem into a series of maximum weight cliques problems and then we use relaxation labeling to find an approximate solution.

In a set of experiments we apply this algorithm to match shock graphs, a graph representation of the morphological skeleton. The results of these experiments are very encouraging, showing that the algorithm is able to match similar shapes together. Our future plans are twofold. First, we will develop an enhanced version of the method for weighted tree matching. Second, we are planning to investigate whether we can use multidimensional scaling and pairwise clustering methods to induce perceptually meaningful equivalence classes on the different shapes used in our study.
















														
1.000	1.000	1.000	1.000	0.944	1.000	1.000	1.000	1.000	0.947	0.947	0.872	1.000	1.000	1.000
0.875	0.875	0.900	0.900	0.944	0.933	0.917	0.875	0.947	0.947	0.872	0.802	0.802	0.840	0.823
0.875	0.875	0.900	0.900	0.944	0.933	0.917	0.875	0.842	0.842	0.821	0.586	0.586	0.840	0.823
0.875	0.875	0.900	0.900	0.889	0.867	0.833	0.875	0.632	0.632	0.564	0.517	0.517	0.538	0.538
1.000	1.000	1.000	1.000	0.794	0.825	0.900	1.000	0.947	0.947	0.872	0.766	0.766	0.774	0.774
0.875	0.875	0.900	0.900	0.733	0.825	0.750	0.875	0.895	0.947	0.821	0.584	0.584	0.820	0.820
0.875	0.875	0.900	0.900	0.794	0.825	0.825	0.875	0.842	0.842	0.821	0.650	0.650	0.638	0.638
0.875	0.875	0.900	0.900	0.794	0.825	0.600	0.875	0.632	0.632	0.564	0.438	0.438	0.455	0.455
1.000	1.000	0.900	0.900	0.778	0.794	0.825	1.000	0.872	0.872	0.850	0.765	0.765	0.774	0.774
0.875	1.000	0.788	0.788	0.622	0.794	0.825	0.875	0.872	0.872	0.850	0.540	0.540	0.683	0.683
0.875	0.875	0.675	0.788	0.700	0.794	0.825	0.875	0.821	0.821	0.800	0.630	0.630	0.683	0.683
0.875	0.875	0.675	0.675	0.622	0.794	0.825	0.875	0.610	0.610	0.700	0.540	0.540	0.455	0.455
0.900	0.900	0.900	0.900	0.778	0.758	0.825	0.900	0.774	0.865	0.840	0.740	0.740	0.750	0.750
0.900	0.900	0.788	0.788	0.700	0.657	0.825	0.900	0.683	0.683	0.663	0.610	0.610	0.584	0.584
0.675	0.675	0.788	0.675	0.700	0.657	0.825	0.675	0.683	0.683	0.663	0.653	0.653	0.656	0.656
0.675	0.675	0.788	0.675	0.622	0.607	0.825	0.675	0.455	0.455	0.442	0.430	0.430	0.438	0.438
0.900	0.900	0.900	0.900	0.765	0.758	0.816	0.900	0.763	0.763	0.750	0.708	0.708	0.752	0.752
0.900	0.900	0.788	0.788	0.720	0.642	0.748	0.900	0.687	0.687	0.675	0.708	0.708	0.705	0.705
0.675	0.675	0.788	0.675	0.630	0.642	0.680	0.675	0.687	0.687	0.675	0.657	0.657	0.584	0.584
0.675	0.675	0.675	0.675	0.540	0.583	0.612	0.675	0.687	0.687	0.675	0.607	0.607	0.658	0.658

Fig. 2. Shapes and their top five matches in order of similarity (see text).

References

- [1] H. G. Barrow and R. M. Burstall, Subgraph isomorphism, matching relational structures and maximal cliques, *Inf. Proc. Letters*, Vol. 4, pp.83, 84, 1976.
- [2] I. M. Bomze, M. Pelillo, and V. Stix, Approximating the maximum weight clique using replicator dynamics, *IEEE Transactions on Neural Networks*, Vol. 11, 2000.
- [3] H. Bunke and G. Allermann, Inexact graph matching for structural pattern recognition, *Pattern Recognition Letters*, Vol 1, pp. 245-253, 1983.
- [4] H. Bunke and A. Kandel, Mean and maximum common subgraph of two graphs, *Pattern Recognition Letters*, Vol. 21, pp. 163-168, 2000.
- [5] W. J. Christmas and J. Kittler, Structural matching in computer vision using probabilistic relaxation, *PAMI*, Vol. 17, pp. 749-764, 1995.
- [6] M. A. Eshera and K-S Fu, An image understanding system using attributed symbolic representation and inexact graph-matching, *PAMI*, Vol 8, pp. 604-618, 1986.
- [7] L. E. Gibbons et al. Continuous characterizations of the maximum clique problem, *Math. Oper. Res.*, Vol. 22, pp. 754-768, 1997
- [8] S. Gold and A. Ragaranjan, A graduated assignment algorithm for graph matching, *IEEE PAMI*, Vol. 18, pp. 377-387, 1996.
- [9] T. S. Motzkin and E. G. Straus, Maxima for graphs and a new proof of a theorem of Turán, *Canadian Journal of Mathematics*, Vol. 17, pp. 533-540, 1965.
- [10] M. Pelillo, The dynamics of relaxation labeling process, *J. Math. Imaging Vision*, Vol. 7, pp. 309-323, 1997
- [11] M. Pelillo, K. Siddiqi, and S. W. Zucker, Matching hierarchical structures using association graphs, *IEEE PAMI*, Vol. 21, pp. 1105-1120, 1999.
- [12] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, Indexing using a spectral encoding of topological structure, In *CVPR*, June 1999.
- [13] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker, The hamilton-jacobi skeleton, In *ICCV*, pp. 828-834, September 1999.
- [14] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, Shock graphs and shape matching, *Int. J. of Computer Vision*, Vol. 35, pp. 13-32, 1999.
- [15] K-C Tai, The tree-to-tree correction problem, *Journal of the Association for Computing Machinery*, Vol. 26, pp. 422-433, 1979.
- [16] W. H. Tsai and K. S. Fu, Error-correcting isomorphism of attributed relational graphs for pattern analysis, *Trans. on Sys., Man, and Cyb.*, Vol. 9, 1979.
- [17] J. T. L. Wang, K. Zhang, and G. Chirn, The approximate graph matching problem, In *ICPR*, pp. 284-288, 1994.
- [18] R. C. Wilson and E. R. Hancock, Structural matching by discrete relaxation, *PAMI*, 1997.
- [19] K. Zhang, A constrained edit distance between unordered labeled trees, *Algorithmica*, Vol. 15, pp. 205-222, 1996.
- [20] K. Zhang and D. Shasha, Simple fast algorithms for the editing distance between trees and related problems, *SIAM J. of Computing*, Vol. 18, pp. 1245-1262, 1989.
- [21] K. Zhang, R. Statman, and D. Shasha, On the editing distance between unordered labeled trees, *Information Processing Letters*, Vol. 42, pp. 133-139, 1992.