Shape Matching and Object Recognition

Title:Matching Hierarchical Structures using Association Graphs

Authors: Pelillo, Siddiqi, and Zucker
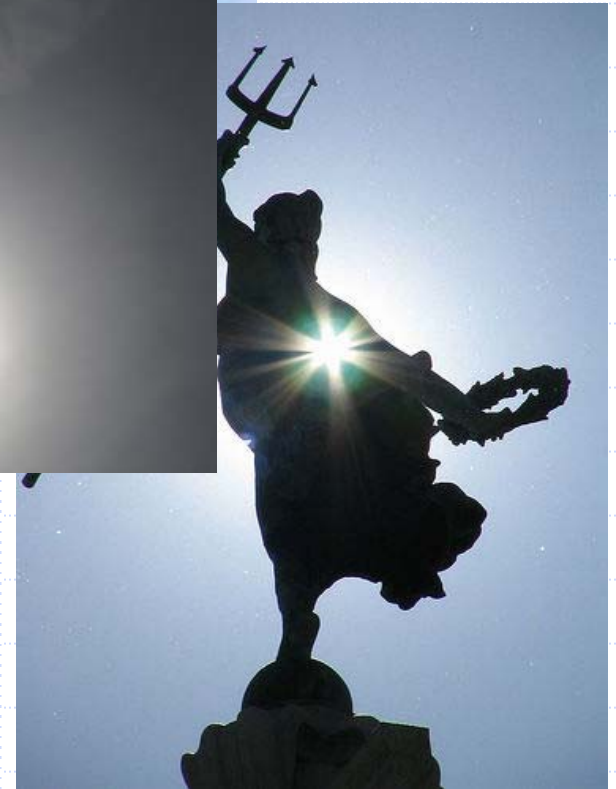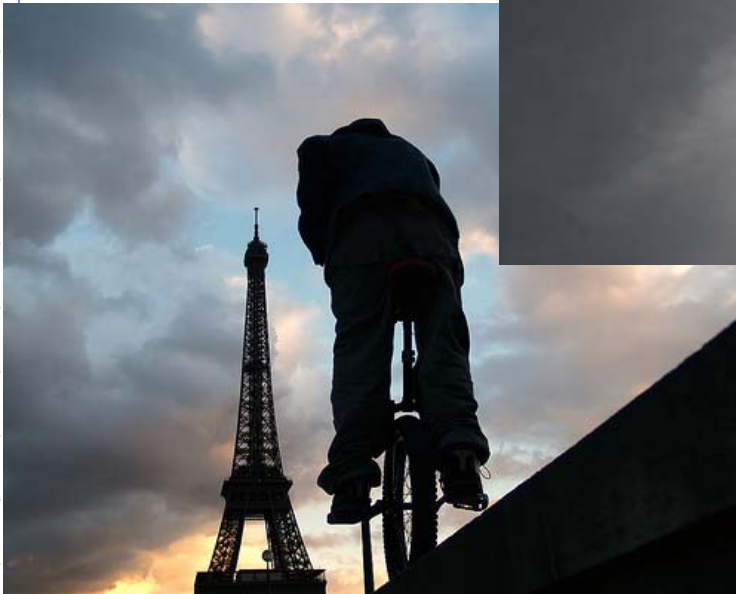
Publication: IEEE PAMI Vol. 21, No. 11
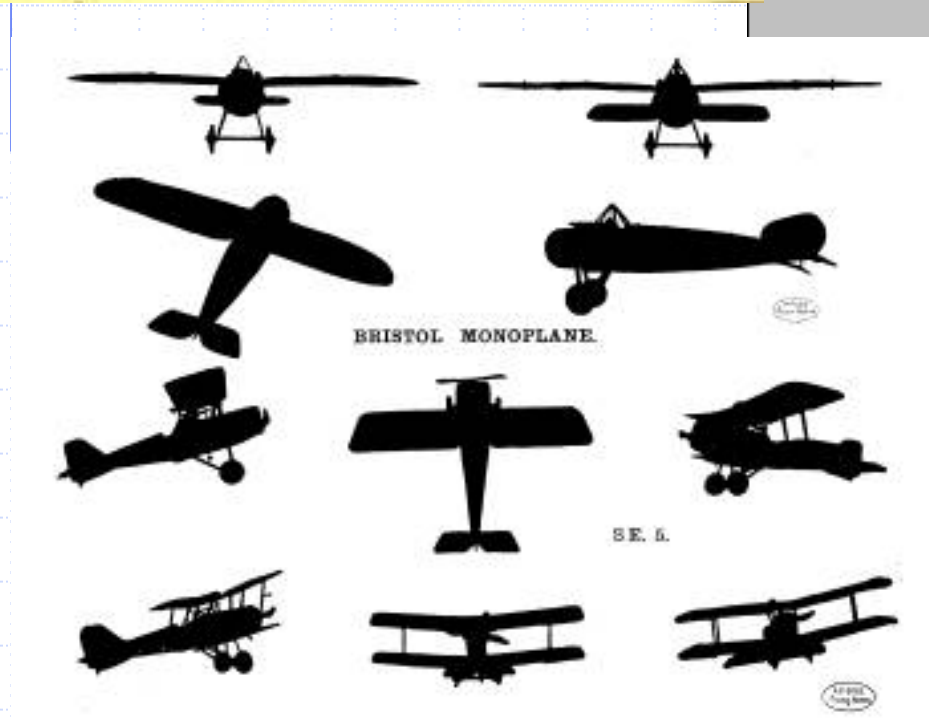
Year: 1999

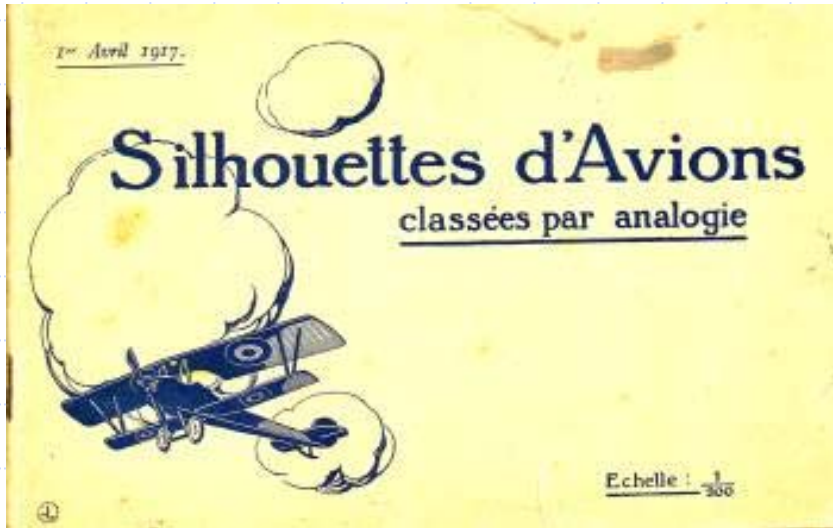URL: http://www.dsi.unive.it/~pelillo/papers/pami99.pdf

Graph Algorithms and Object Recognition: max clique

Rappresentazione e riconoscimento di oggetti

Graph Algorithms and Object Recognition: max clique

Graph Algorithms and Object Recognition: max clique

Graph Algorithms and Object Recognition: max clique

Shock Graph Representation of Shape

• **Shocks** (or medial axis or skeleton) are locus of centers of maximal circles that are bitangent to shape boundary

Shape boundary

Shocks

Real Example
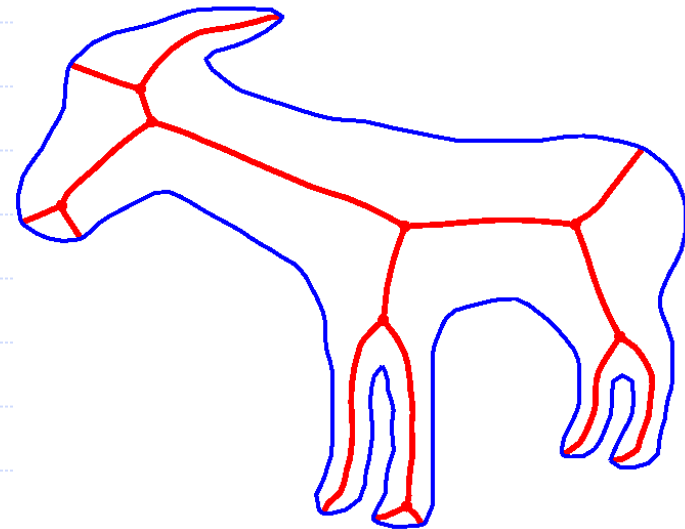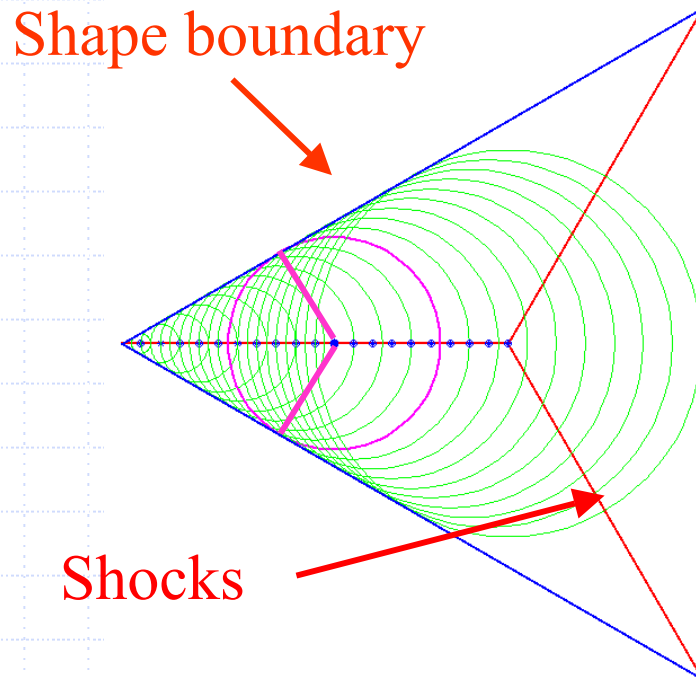
Graph Algorithms and Object Recognition: max clique
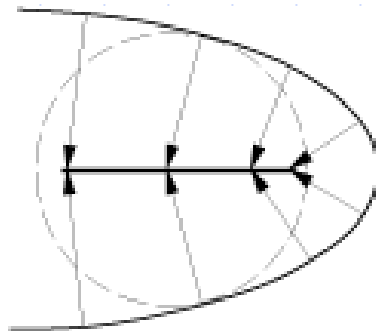
Type 1
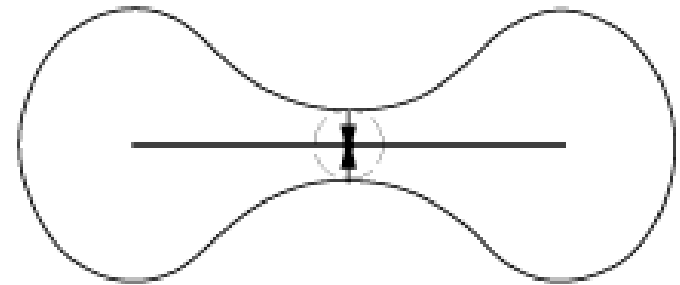
Type 2
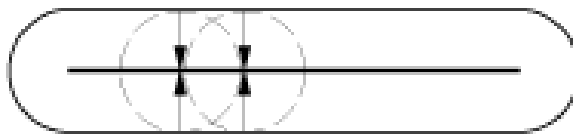
Type 3

Type 4

Shock Categories

Graph Algorithms and Object Recognition: max clique

Graph Algorithms and Object Recognition: max clique
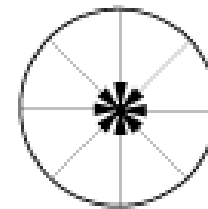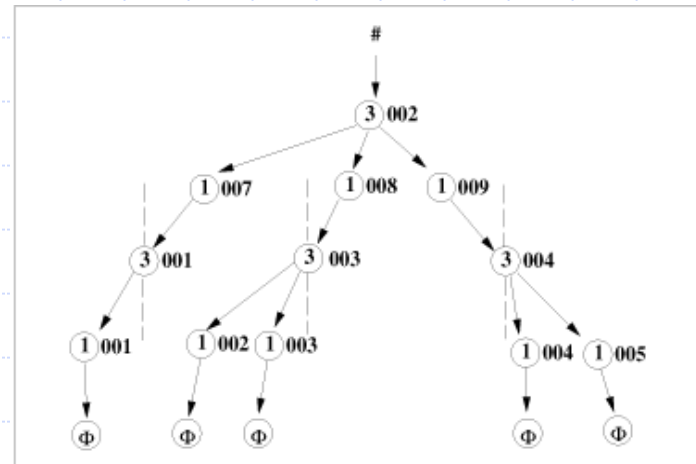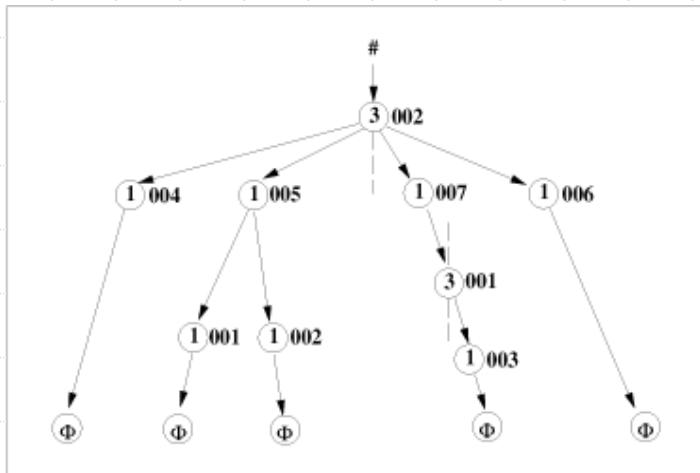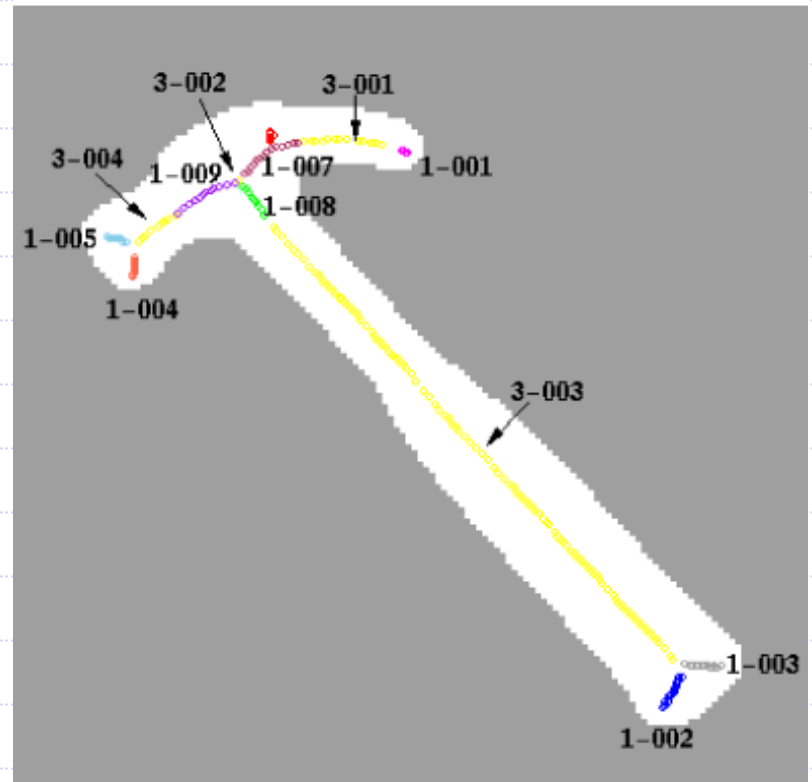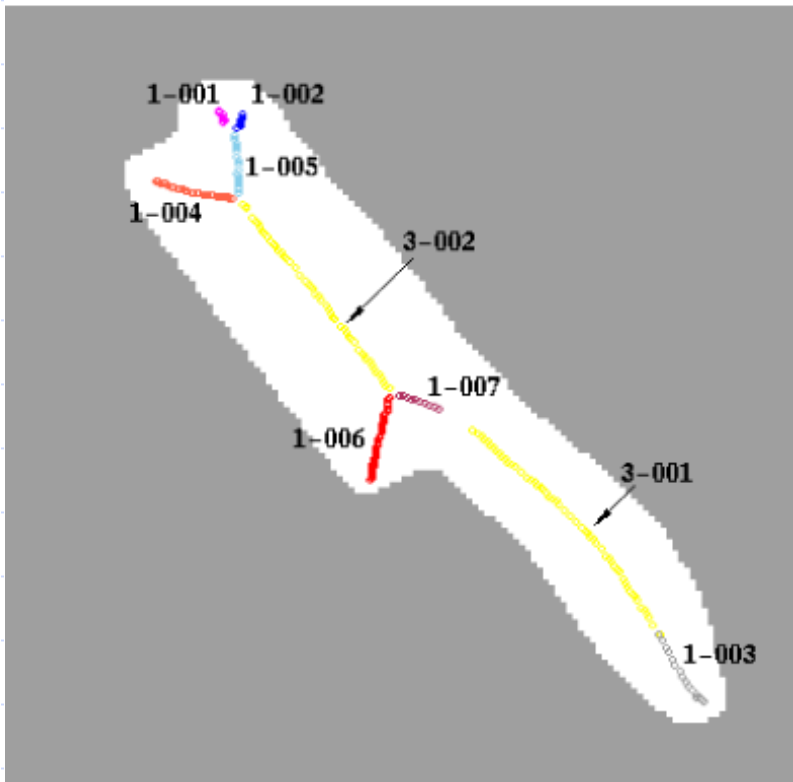
Riconoscimento (matching)

Graph Algorithms and Object Recognition: max clique
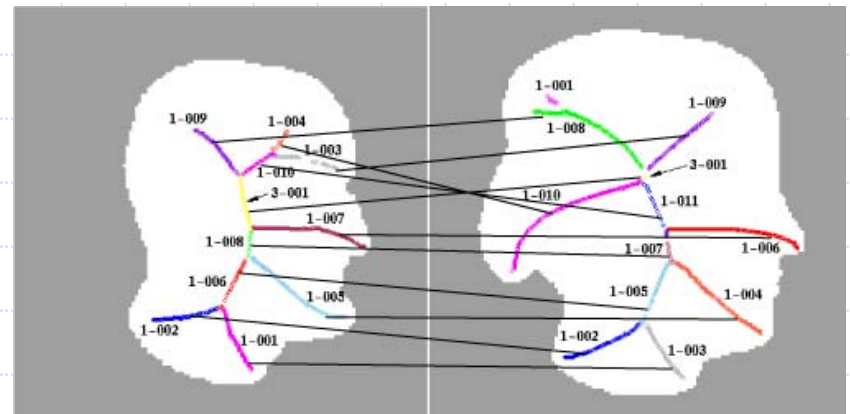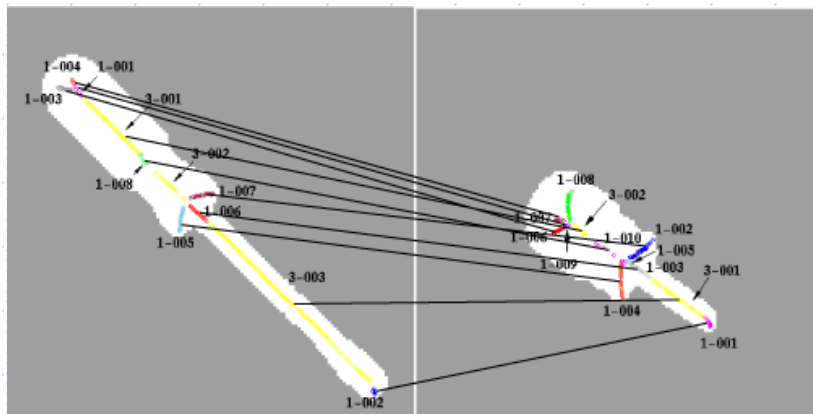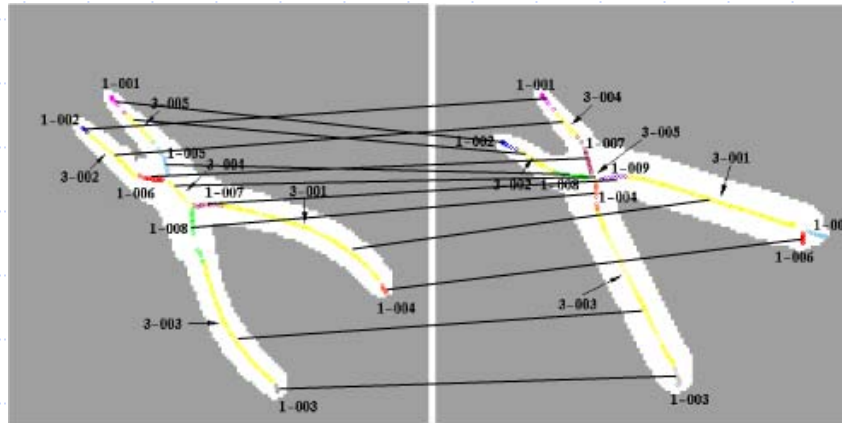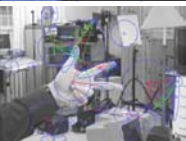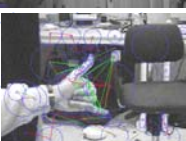
Distance Between Shapes

- **Distance** between shapes is important for recognition

- **Shape space** is the collection of all shapes
    - shape is a point (shock graph) in the space
    - shape deformation (shock graph) sequence is a path through the space

- **Cost of the optimal deformation sequence** is the minimim distance from A to B
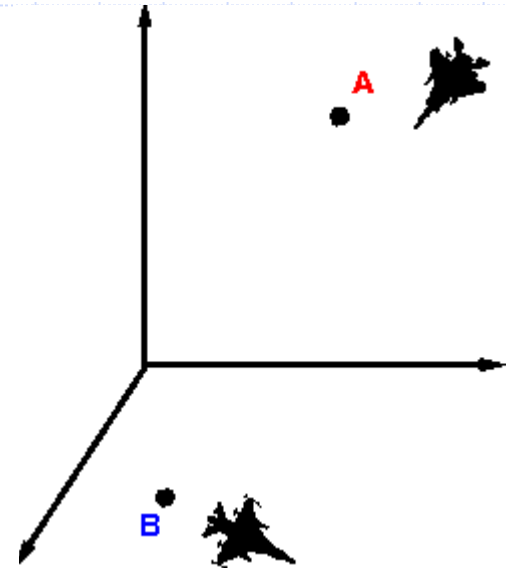
Un'applicazione:
Il riconoscimento di gesti

	12.35	5.73	6.02	7.35	10.25	11.15	9.12
	12.35	5.73	6.02	7.35	10.25	11.15	9.12
	4.05	**6.88**	5.22	3.27	2.84	4.18	5.95
	5.37	3.13	**8.40**	4.47	7.56	4.21	2.72
	15.18	9.02	5.44	13.19	10.18	**15.95**	13.22
	21.84	11.01	12.17	15.88	9.21	17.75	16.37
	10.43	3.41	4.19	4.00	7.26	5.69	4.96

Graph Algorithms and Object Recognition: max clique

Organizzazione di database visuali
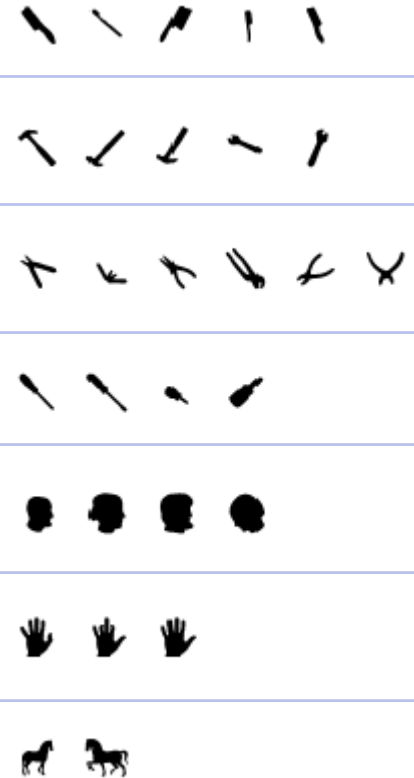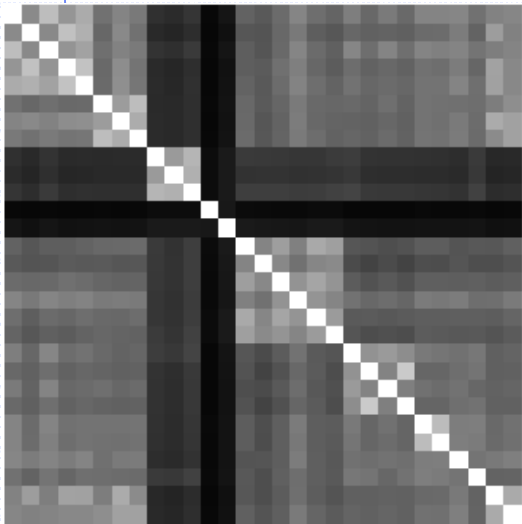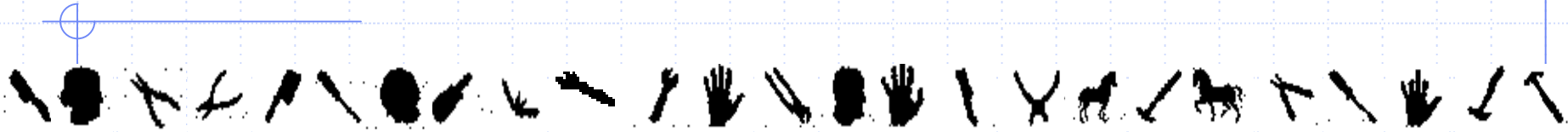
Motivation

◆ Many hierarchical structures in computer vision can be represented as trees.

◆ Matching two hierarchical structures can therefore be formulated as finding their largest isomorphic subtree.

◆ Equivalently, we can find the maximum clique in their association graph, effectively reducing a hierarchical matching problem to a "flat," discrete optimization problem.

◆ Powerful, continuous optimization methods applicable only to "flat" problems can be used to find a solution that obeys hierarchical constraints.

Graph Algorithms and Object Recognition: max clique

Outline

◆ Subtree isomorphism

◆ An equivalent, maximum clique formulation

◆ An equivalent, continuous, quadratic optimization formulation

◆ A dynamical systems solution framework

◆ An extension to free trees

◆ An extension to weighted maximum clique

◆ Conclusions

Preliminaries

Given a graph $G=(V, E)$, with V the set of nodes, and E the set of edges:

◆ Two nodes $u,v \in V$ are *adjacent* (denoted $u \sim v$) if they are connected by an edge.

◆ A *path* is any sequence of distinct nodes $u_0 \, u_1 \, \cdots \, u_n$ such that $u_{i-1} \sim u_i$, $i=1 \cdots n$.

◆ If $u_0=u_n$, the path is a *cycle*.

◆ A graph is *connected* if any pair of nodes is joined by a path.

◆ The *distance* between two nodes u and v is the length of the shortest path joining them.

Graph Algorithms and Object Recognition: max clique

Trees

◆ A *tree* is a connected graph with no cycles.

◆ A *rooted* tree has a distinguished node, the root.

◆ The *level* of a node u in a rooted tree (denoted $lev(u)$) is the distance between u and the root.

◆ If $u \quad v$ and $lev(v) - lev(u) = 1$, u is the *parent* of v and, conversely, v is a child of u.

Property: In a tree, any two nodes are connected by a *unique* path.

Subtree Isomorphism

Let $T_1=(V_1, E_1)$ and $T_2=(V_2, E_2)$ be two rooted 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, given $u,v \in H_1$, we have:

1. $u \prec v \Leftrightarrow \phi(u) \prec \phi(v)$
2. u is v's parent $\Leftrightarrow$ $\phi(u)$ is $\phi(v)$'s parent.
3. the induced subgraphs are connected.

16

A subtree isomorphism is maximal if there is no other subtree isomorphism $\phi' : H_1' \rightarrow H_2'$ with $H_1 \quad H_1'$, 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.

Graph Algorithms and Object Recognition: max clique

Subgraph Isomorphism as a Maximum Clique Problem
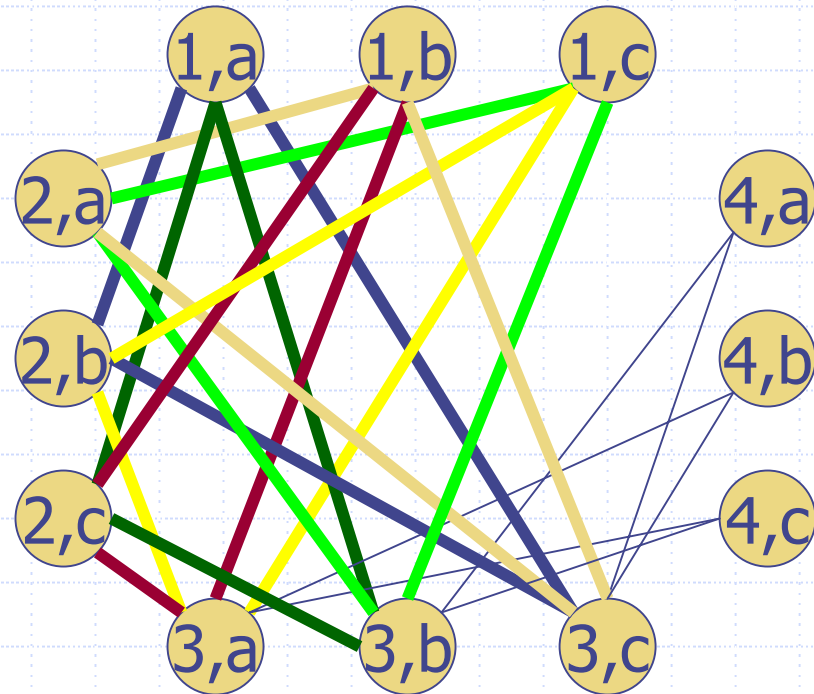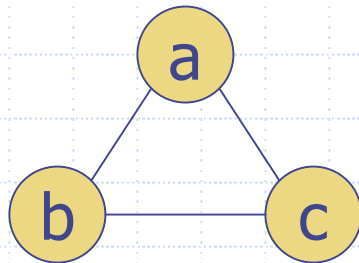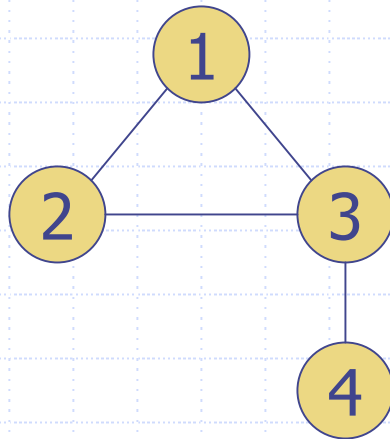
$(u,w) \sim (v,z)$ iff $(u \sim v$ AND $w \sim z)$ OR $(u \, !\sim v$ AND $w \, !\sim z)$

Graph Algorithms and Object Recognition: max clique

The Loss of Hierarchical Structure

In the standard formulation of the association graph, the solutions may not preserve the hierarchical structure of the trees being matched.

Example:
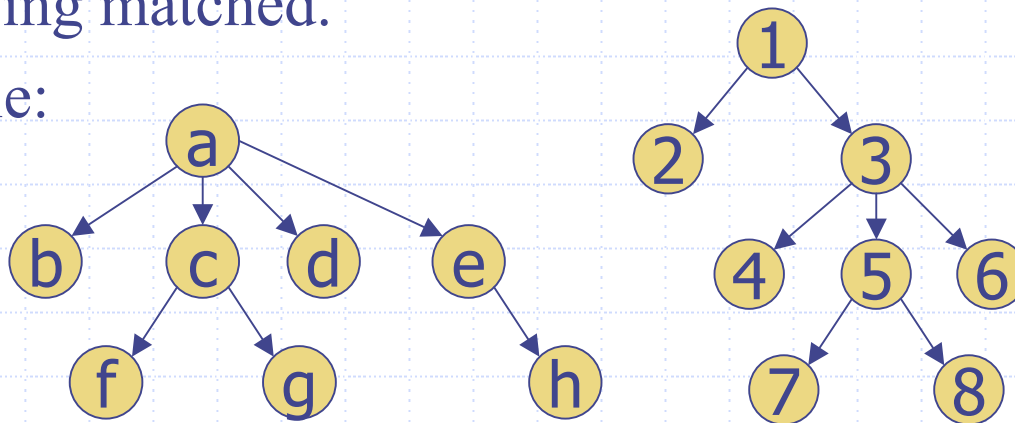
Using the standard association graph, the assignment 2 → h will also be included in the final solution, but the result violates hierarchical constraints, and is also not a tree.

Graph Algorithms and Object Recognition: max clique

Solution: Path Strings

Let u and v be two distinct nodes of a rooted tree, with $u=x_0x_1\cdots x_n=v$ the (unique) path joining them.

The path-string of u and v, denoted by $str(u,v)$, is the string $s_1s_2\cdots s_n$ on the alphabet $\{-1,+1\}$ where $s_i=lev(x_i)-lev(x_{i-1})$, $i=1\cdots n$.

Example:
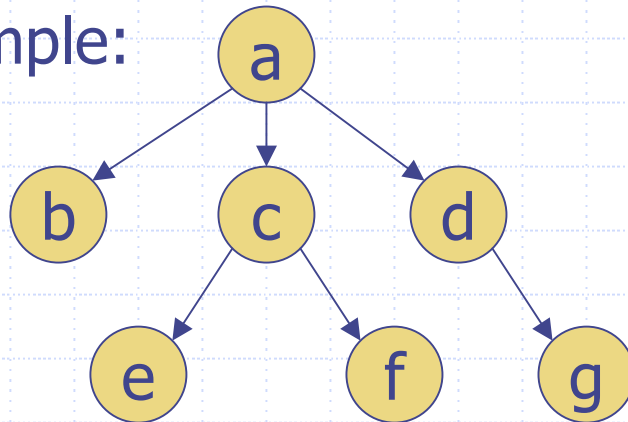
$str(e,g)=-1-1+1+1$
$str(u,u)=\varepsilon$, the null string

Graph Algorithms and Object Recognition: max clique

Deriving the Tree Association Graph (TAG)

The TAG of two attributed trees $T_1=(V_1,E_1)$ and $T_2=(V_2,E_2)$ is the (weighted) graph $G=(V,E)$ where:

1. $V=V_1 \times V_2$

2. for any two nodes (u,w) and (v,z) in V:
$$(u,v) \quad (w,z) \quad str(u,w)=str(v,z)$$

The Main Theorem

A subset of vertices of a graph $G = (V,E)$ is said to be a *clique* if all its nodes are mutually adjacent.

A maximal clique in G is one which is not contained in any other clique having larger size.

Theorem 1 Any maximal (maximum) subtree isomorphism between two rooted trees induces a maximal (maximum) clique in the corresponding TAG and vice versa. (proof in paper)

Graph Algorithms and Object Recognition: max clique

Why Association Graphs for Subtree Isomorphism?

The association graph approach is attractive for several reasons:

• it allows us to cast the relational structure matching problem in terms of a well-known problem in graph theory, for which both theory and algorithms are available.

• many powerful heuristics for solving the maximimum clique problem are available (Bomze et al, 1999).

• it generalizes to many-to-many matching problems (see Pelillo et al., 1999 (IWVF), for a review)

Graph Algorithms and Object Recognition: max clique

Some Notation

Given an arbitrary weighted graph *G(V,E,ω)* with *n* nodes: 1

- *S$_n$* is the standard simplex in R^n:

$$S_n = \left\{ x \in \Re^n : \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0, \forall i \right\}$$

1

1

1
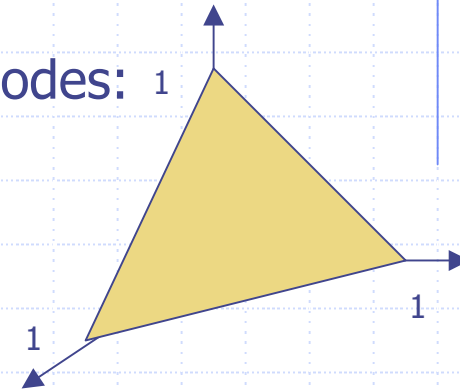
- If *C* ⊆ *G*, **x**^{c} will denote its characteristic vector, which is defined as:

$$x_i^c = \left\{ \begin{array}{ll} 1/|C|, & \text{if } u_i \in C \\ 0, & \text{otherwise} \end{array} \right\}$$

- where |*C*| denotes the cardinality of *C*.

Graph Algorithms and Object Recognition: max clique

A Continuous Formulation for MAX-CLIQUE

Consider the following quadratic function:

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

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

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

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

The Motzkin-Straus theorem (1965) establishes a connection between the global (local) maximizers of f in S_n and maximum (maximal) cliques of G.

Namely, a subset of vertices C of a graph G is a maximum clique if and only if its characteristic vector x^c is a global maximizer of f on S_n.

This allows us to shift to the continuous domain, drawing on continuous optimization techniques to solve the corresponding discrete problem.

Graph Algorithms and Object Recognition: max clique

Problem

- Unfortunately, there is a problem with the Motzkin-Strauss formulation.

- Namely, spurious solutions exist that do not obey the properties of a characteristic vector, i.e., the solution doesn't converge to the barycenter of a simplex face.

- In these cases, the cardinality of the solution is known but the assignments are not.

Graph Algorithms and Object Recognition: max clique

Solution

Bomze has recently introduced a solution using a regularized version of f:

$$g(x) = x'Ax + \frac{1}{2} x'x$$

which is obtained by substituting in f the following adjacency matrix:

$$\hat{A}(x) = A + \frac{1}{2} I_n$$

where I_n is the $n \times n$ identity matrix.

Graph Algorithms and Object Recognition: max clique

A Stronger Result

Theorem 2

Let $C \subseteq V$, and let x^c be its characteristic vector. Then:

- C is a maximum clique of G $\Leftrightarrow$ x^c is a global maximizer of g in S_n. In this case, $|C| = 1/2(1 - g(\boldsymbol{x}^c))$.

- C is a maximal clique of G $\Leftrightarrow$ x^c is a local maximizer of g in S_n.

- All local (and hence global) maximizers of g on S_n are strict.

See (Bomze, Pelillo, and Stix, 1999) for proof.

Graph Algorithms and Object Recognition: max clique

An Evolutionary Approach to Solving the Continuous Optimization Problem

The search for an optimal solution on the simplex can be modeled as the evolution of a set of populations over time.

Following (Hofbauer & Sigmund, 1988), the components of our vector **x** represent the frequencies of the various populations.

The rate of increase of a particular population E_i equals the increase in fitness of the population over the average fitness:

$$\frac{\dot{x}_i}{x_i} = \text{fitness of } E_i - \text{average fitness}$$

Graph Algorithms and Object Recognition: max clique

$$\frac{\dot{x}_i}{x_i} = \text{fitness of } E_i - \text{average fitness}$$

$$\dot{x}_i = x_i(f_i(\mathbf{x}) - \bar{f}(\mathbf{x})), \quad i = 1, \dots, n$$

This is called the replicator equation, and it models the evolution of behaviour in animal conflicts.

The trajectory of the populations, including the solution, is guaranteed to stay on the surface of the simplex.

Graph Algorithms and Object Recognition: max clique

If f is linear, i.e., f is an $n \times n$ matrix such that $f_i(\mathbf{x}) = (W\mathbf{x})_i$, then we can write:

$$\dot{x}_i(t) = x_i(t)(f_i(\mathbf{x}(t)) - \bar{f}(\mathbf{x}(t))), \quad i = 1, \ldots, n$$

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

and in the discrete domain:

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

Graph Algorithms and Object Recognition: max clique

The Fundamental Theorem of Natural Selection

Theorem 3 If W=W', then the function

$$f(x) = x'Wx$$

is strictly increasing along any non-stationary trajectory of both continuous-time and discrete-time replicator dynamics. In other words, $\forall t \geq 0$:

$$\frac{d}{dt} f(x(t)) > 0$$

for the contonuous-time dynamics, and

$$f(x(t+1)) > f(x(t))$$

for the discrete-time dynamics, unless *x(t)* is a stationary point. Furthermore, any such trajectory converges to a stationary point (see Fisher (1930), Crow and Kimura (1970), Hofbauer & Sigmund (1988)).

Graph Algorithms and Object Recognition: max clique

Back to Tree Matching

Let T_1 and T_2 be two attributed trees, $G=(V,E)$ be the corresponding TAG, and define

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

our regularized Motzkin-Strauss formulation.

The replicator dynamical system starting from an arbitrary initial state (typically, the barycenter of S_i) will eventually converge to a maximizer of $x'Wx$ over the standard simplex.

This will correspond to a maximal clique in the TAG, and hence to a maximal subtree isomorphism between T_1 and T_2.

Graph Algorithms and Object Recognition: max clique

Shock Trees
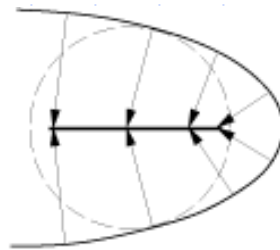
Type 1
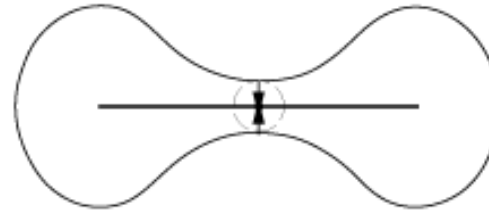
Type 2

Type 3

Type 4

Shock Graph
Siddiqi, Shokoufandeh, Dickinson, and Zucker, 1999

Graph Algorithms and Object Recognition: max clique

Graph Algorithms and Object Recognition: max clique

Results

Graph Algorithms and Object Recognition: max clique

Matching Free Trees

Title: Matching Free Trees, Maximal Cliques, and Monotone Game Dynamics

Author: M. Pelillo

Publication: IEEE PAMI

Year: 2002

URL: http://www.dsi.unive.it/~pelillo/papers/pami-2001.ps.gz

Extension to Free Trees

A free tree (or simply a tree) is a connected graph with no cycles.

Note: Unlike rooted trees, free trees have no distinguished node playing the role of root. Hence, no hierarchy is imposed on the tree.

Property: In a tree, any two nodes are connected by a unique path.

Graph Algorithms and Object Recognition: max clique

Deriving the Free Tree Association Graph (FTAG)

The *free tree association graph* (FTAG) of two attributed trees $T_1=(V_1,E_1)$ and $T_2=(V_2,E_2)$ is the (weighted) graph $G=(V,E)$ where:

1. $V=V_1 \times V_2$

2. for any two nodes (u,w) and (v,z) in V:

$$(u,v) \quad (w,z) \quad d(u,w)=d(v,z)$$

Graph Algorithms and Object Recognition: max clique

The Main Theorem Holds for Free Trees

Theorem 1 Any maximal (maximum) similarity subtree isomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted FTAG and vice versa. (proof in paper)

Faster Replicator Dynamics

Continuous-time version (κ is a positive constant):

$$\frac{d}{dt}x_i(t) = x_i(t)\left(\frac{e^{\kappa\pi_i(t)}}{\sum_{j=1}^{n}x_j(t)e^{\kappa\pi_j(t)}} - 1\right)$$

Discrete-time version:

$$x_i(t+1) = \frac{x_i(t)e^{\kappa\pi_i(t)}}{\sum_{j=1}^{n}x_j(t)e^{\kappa\pi_j(t)}}$$

Graph Algorithms and Object Recognition: max clique

Tree Matching Replicator Equations

Let T' and T'' be two free trees, let A_G denote the adjacency matrix of the coresponding FTAG, and let

$$W = A_G + \frac{1}{2} I \quad (= W' \geq 0),$$

where I is the identity matrix.

The replicator systems, starting from an arbitrary initial state, will eventually converge to a maximizer of the function

$$f_G(x) = x'Ax + \frac{1}{2} x'x$$

over the simplex. This will correspond to a maximal clique of the FTAG, and hence to a "maximal" subtree isomorphism between T' and T''.

Graph Algorithms and Object Recognition: max clique

Matching Shape Axis Trees

Recently, Liu, Geiger, and Kohn (1998) introduced a new representation for shape based on the idea of self-similarity.

Given a closed planar shape, they consider two different parameterizations of its contour, one oriented counterclockwise:

$$\Gamma(s) = \left\{ x(s) : 0 \le s \le 1 \right\}$$

and the other clockwise:

$$\hat{\Gamma}(t) = \left\{ \hat{x}(t) = x(1-t) : 0 \le t \le 1 \right\}$$

Graph Algorithms and Object Recognition: max clique

• By minimizing an appropriate cost functional, they find a "good" match between the two parameterizations.

• The shape axis (SA) is defined as the loci of middle points between the matched contour points.

• From a given SA, it is possible to construct a unique free tree, called the SA-tree, by grouping the discontinuities contained in the SA.

Graph Algorithms and Object Recognition: max clique

Results
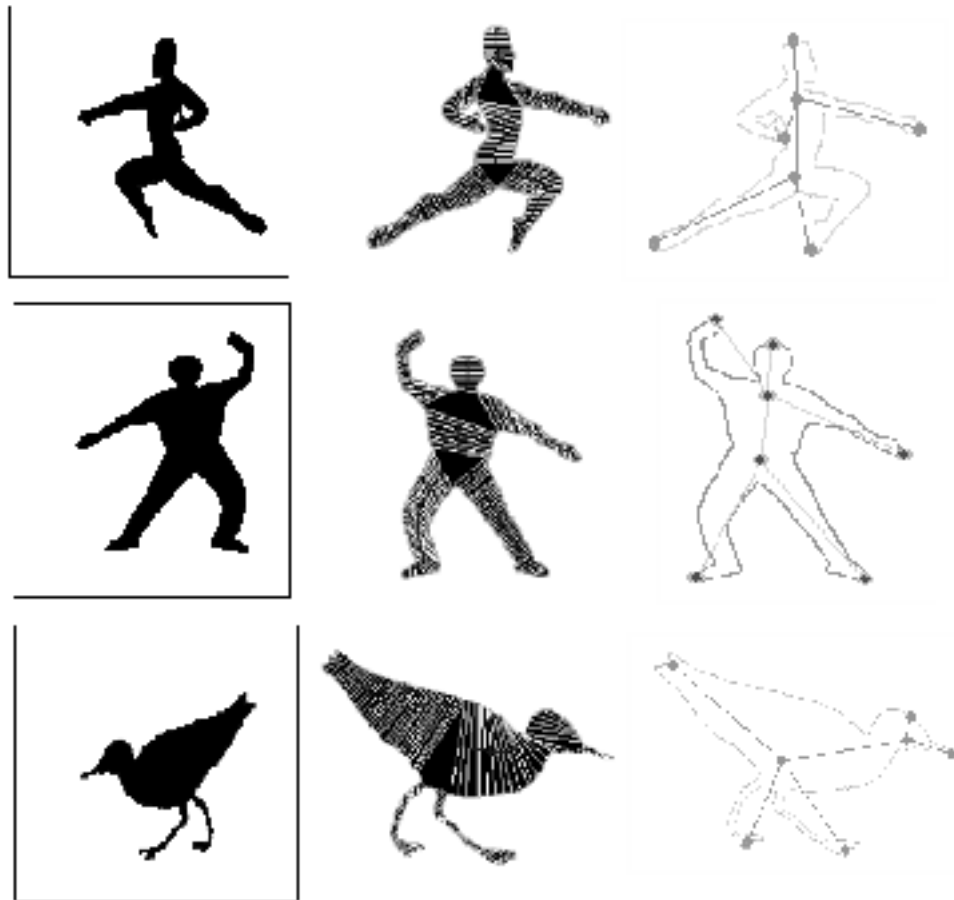
SA-tree construction: input shapes (left); shape-axis model (middle); SA-trees (right)

Graph Algorithms and Object Recognition: max clique
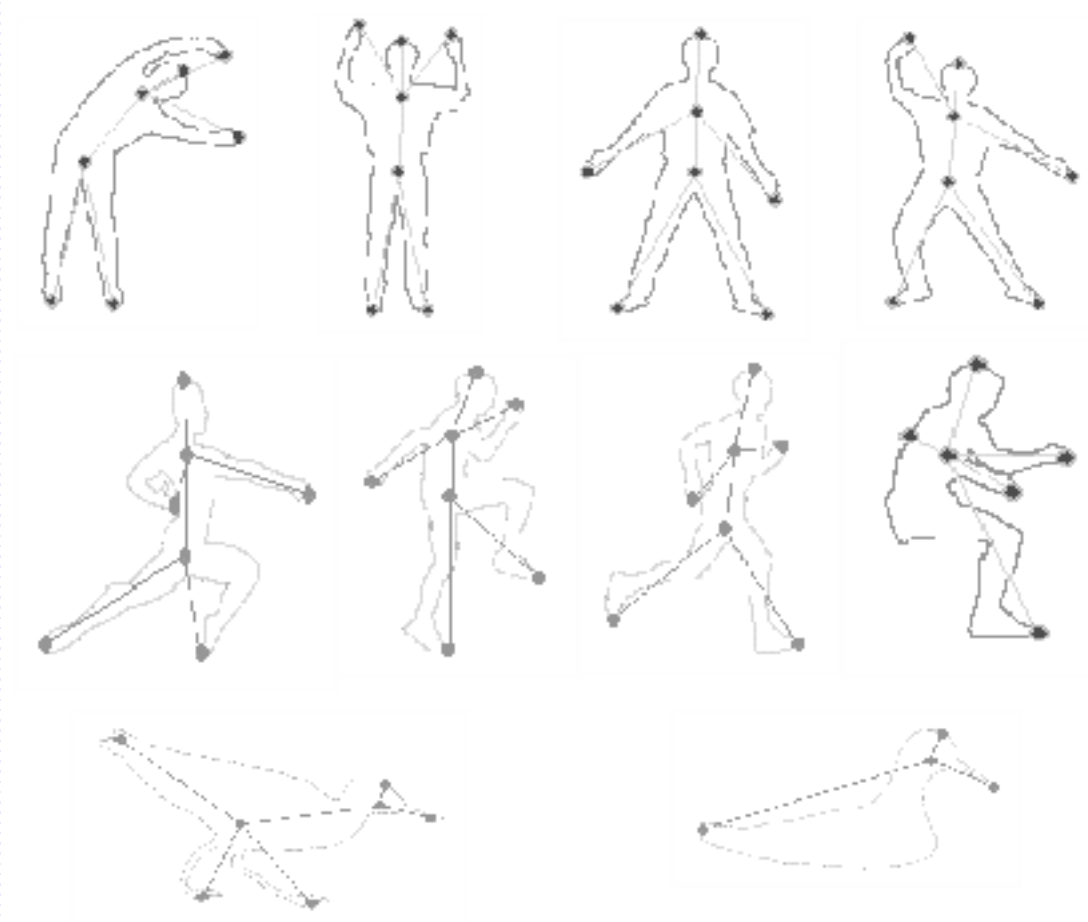
Examples of SA-trees, under various shape deformations.

Graph Algorithms and Object Recognition: max clique

Algorithm was tested on a selection of 17 shapes (SA-trees) representing six different object classes (horse, human, bird, dog, sheep, and rhino).

Both the discrete-time first-order dynamics and its exponential counterpart (with $\kappa=10$) were used.

The algorithms were started from the simplex barycenter and stopped when either a maximal clique was found, or the distance between two successive points was smaller than a fixed threshold.

Each shape was matched to each other shape (including itself), and in all 289 trials, both algorithms returned the maximum isomorphism, i.e., a maximum clique in the FTAG.

Graph Algorithms and Object Recognition: max clique
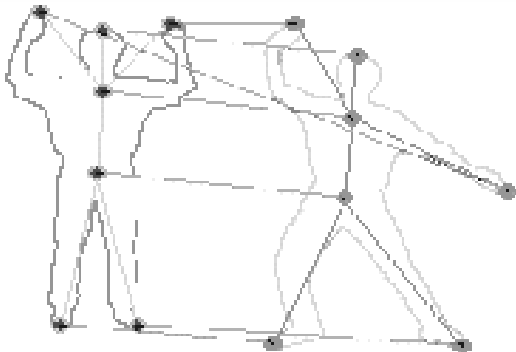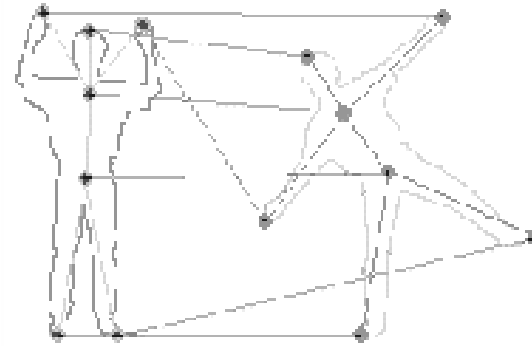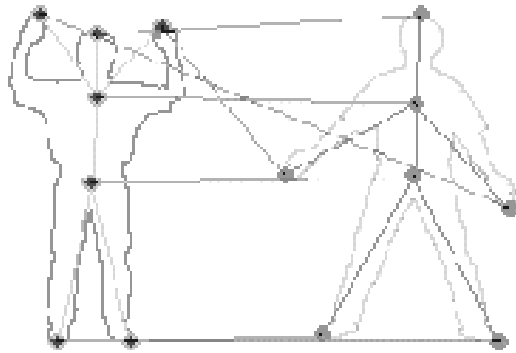
SA-tree matching examples.

Graph Algorithms and Object Recognition: max clique

Matching (Larger) Random Trees

A hundred 100-node free trees were generated uniformly at random using a procedure described by Wilf (1981).

Each tree was subject to a corruption process which consisted of randomly deleting a fraction of its terminal nodes.

Corruption levels: 2%, 10%, 20%, 30%, 40%.

Overall, therefore, 500 pairs of trees were obtained.

Both linear and the exponential dynamics were used.

After convergence, the proportion of matched nodes was calculated and averaged.

Graph Algorithms and Object Recognition: max clique

Results obtained over 100-node random trees with various levels of coruption, using the first-order dynamics. Top: percentage of correct matches; Bottom: average computational time taken by the replicator equations.

Graph Algorithms and Object Recognition: max clique
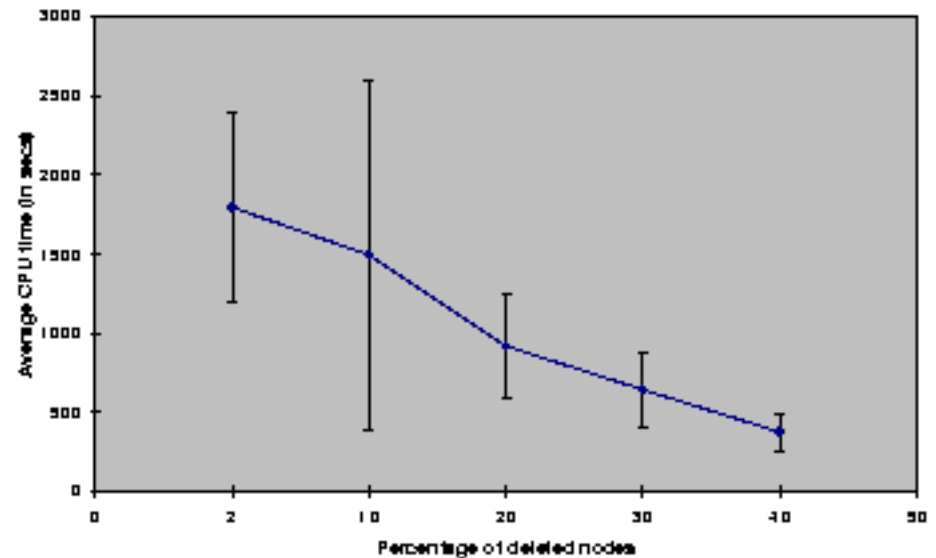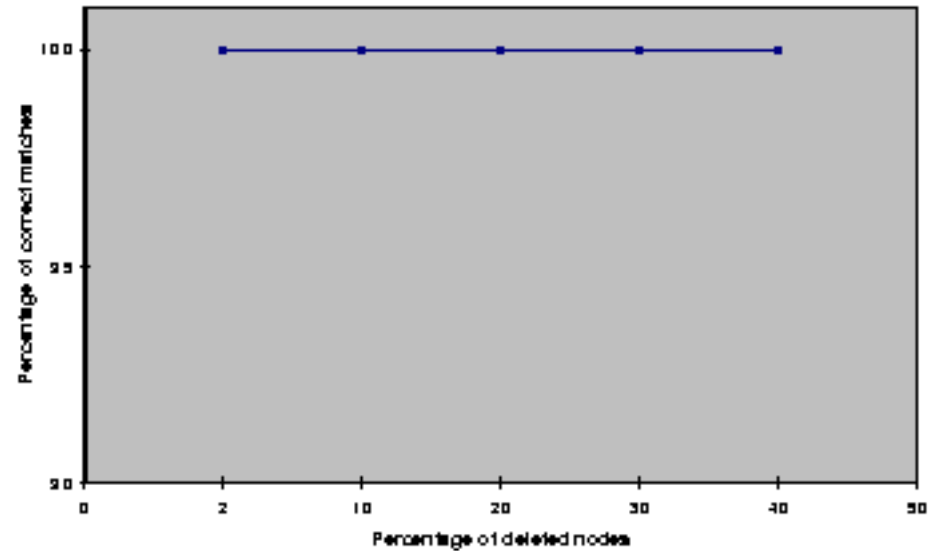
Results obtained over 100-node random trees with various levels of coruption, using the exponential dynamics. Top: percentage of correct matches; Bottom: average computational time taken by the replicator equations.

52
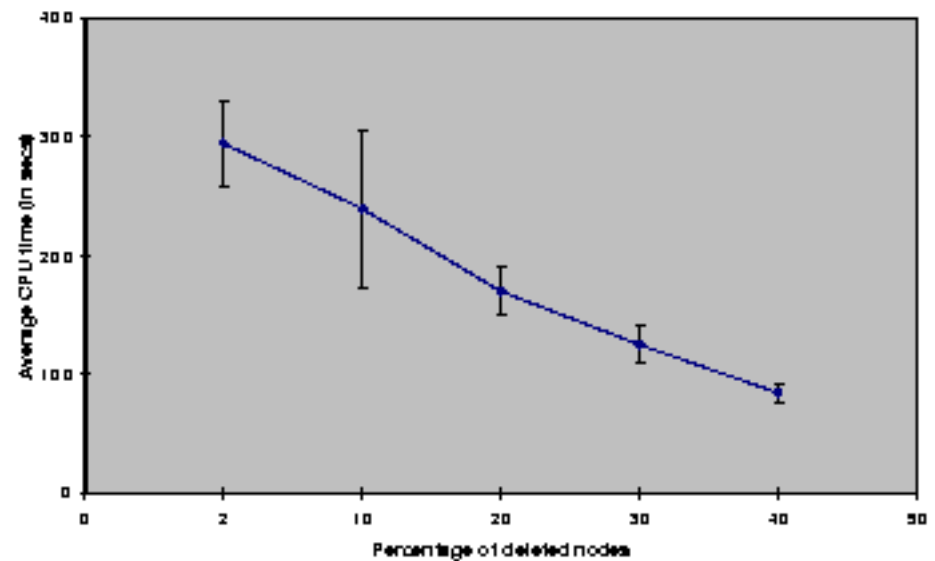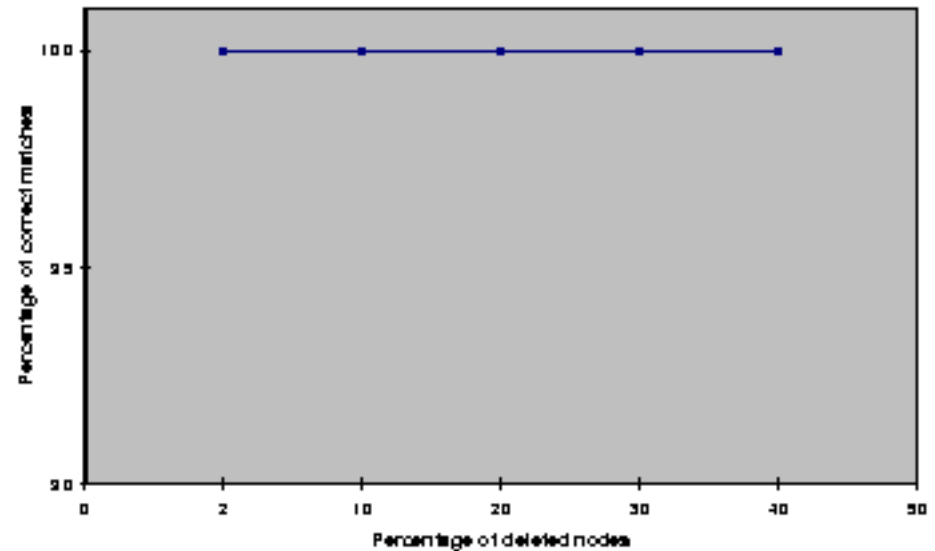
Application: Matching Attributed Hierarchical Structures

Title:Matching Hierarchical Structures using Association Graphs

Authors: Pelillo, Siddiqi, and Zucker

Publication: IEEE PAMI Vol. 21, No. 11

Year: 1999

URL:
http://www.dsi.unive.it/~pelillo/papers/pami99.pdf

Attributed Tree Matching

Let σ be any similarity measure on the attribute space.

An *attributed tree* is a triple $T=(V,E,\alpha)$, where (V,E) is the "underlying" rooted tree and α is a function which assigns an attribute vector $\alpha(u)$ to each node $u \quad V$.

If $\phi : H_1 \rightarrow H_2$ is a subtree isomorphism between two attributed trees $T_1 = (V_1, E_1, \alpha_1)$ and $T_2 = (V_2, E_2, \alpha_2)$, the overall similarity between the induced subtrees $T_1[H_1]$ and $T_2[H_2]$ is defined as:

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

54

The isomorphism ϕ is called a *maximal similarity subtree isomorphism* if there is no other subtree isomorphism $\phi' : H_1' \rightarrow H_2'$ 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.

Graph Algorithms and Object Recognition: max clique

Deriving the Attributed Tree Association Graph (TAG)

The TAG 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:

1. $V=V_1 \times V_2$

2. for any two nodes (u,w) and (v,z) in V:
   $$(u,v) \quad (v,z) \quad str(u,v)=str(w,z)$$

3. for any node (u,w) in V:
   $$\omega(u,w)=\sigma(\alpha_1(u),\alpha_2(w))$$

The Main Theorem holds for Attributed Trees

A subset of vertices of a weighted graph $G=(V,E,\omega)$ is said to be a *clique* if all its nodes are mutually adjacent.

Given a subset of nodes C of V, the total weight assigned to C is the sum of all the weights associated with its nodes.

A maximal weight clique in G is one which is not contained in any other clique having larger total weight.

A maximum weight clique is a clique having largest total weight.

Theorem 1 Any maximal (maximum) similarity subtree isomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding weighted TAG and vice versa. (proof in paper)

Graph Algorithms and Object Recognition: max clique

Some Notation

Given an arbitrary weighted graph $G(V,E,\omega)$ with n nodes: 1

- S_n is the standard simplex in R^n:

$$S_n = \left\{ x \in \mathfrak{R}^n : \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0, \forall i \right\}$$
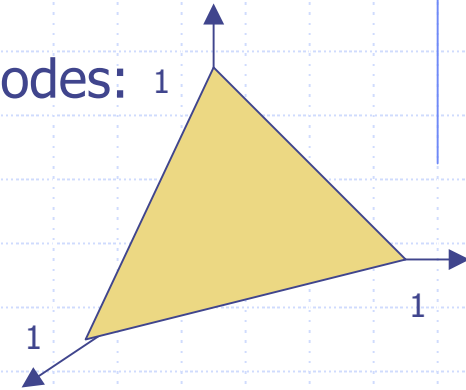
1   1

- If $C \subseteq G$, $\boldsymbol{x}^c$ will denote its characteristic vector, which is defined as:

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

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

Continuous Formulation of MAX-WEIGHT-CLIQUE

Given a weighted graph $G=(V,E,\omega)$, let $M(G)$ be the class of nxn symmetric matrices:

$$B \in M(G) \Leftrightarrow \begin{cases} b_{ii} = \dfrac{1}{2\omega(u_i)}, \forall i = 1\ldots n \\ b_{ij} = 0, \text{ if } u_i \, \square \, u_j \\ b_{ij} = b_{ji} \ge b_{ii} + b_{jj}, \text{ otherwise} \end{cases}$$

Graph Algorithms and Object Recognition: max clique

Weighted Clique/Minimizer Duality

Theorem 2 (extended) Let $G=(V,E,\omega)$ be an arbitrary weighted graph, and let $B \in M(G)$. Then:

• A vector $\boldsymbol{x} \in S_n$ is a local minimizer of $g(\boldsymbol{x})=\boldsymbol{x}'B\boldsymbol{x}$ on S_n iff $\boldsymbol{x} = \boldsymbol{x}^c$, where C is a maximal weight clique of G.

• A vector $\boldsymbol{x} \in S_n$ is a global minimizer of $g(\boldsymbol{x})=\boldsymbol{x}'B\boldsymbol{x}$ on S_n iff $\boldsymbol{x} = \boldsymbol{x}^c$, where C is a maximum weight clique of G.

• All local (and hence global) minimizers of $g(\boldsymbol{x})=\boldsymbol{x}'B\boldsymbol{x}$ on S_n are strict.

See (Bomze, Pelillo, and Stix, 1999) for proof.

M(G) is isomorphic to the positive orthant in

$$\binom{n}{2} - |E| \quad \text{dimensions}$$

This class is a polyhedral pointed cone whose apex is given by the following matrix, the one used in the experiments:

$$b_{ij} = \begin{cases} \dfrac{1}{2\omega(u_i)} & \text{if } i = j \\ 0 & \text{if } i \neq j \text{ and } u_i \; \square \; u_j \\ \dfrac{1}{2\omega(u_i)} + \dfrac{1}{2\omega(u_j)} & \text{otherwise} \end{cases}$$

Graph Algorithms and Object Recognition: max clique

Attributed Tree Matching

Let T_1 and T_2 be two attributed trees, $G=(V,E,\omega)$ be the corresponding weighted TAG, and define

$$W = \gamma \mathbf{ee}' - B$$

where $b=(b_{ij})$ is any matrix in the class $M(G)$, $\mathbf{e}$ is the vector of 1's, and $\gamma = max\ b_{ij}$. Note that since B implies a minimization problem, we transform it to W, so that we can apply our replicator equation maximization framework.

The replicator dynamical system, starting from an arbitrary initial state, will eventually converge to a maximizer of $x'Wx$ (and hence a minimizer of $x'Bx$) over the standard simplex.

The solution yields a maximal weight clique in the weighted TAG, and hence to a maximal similarity subtree isomorphism between T_1 and T_2.

Graph Algorithms and Object Recognition: max clique

Attributed Shock Trees

The vector of attributes assigned to each node $u \in V$ of the attributed shock tree $T=(V,E,\alpha)$ is given by

$$\alpha(u) = (x_1, y_1, r_1, v_1, \theta_1; \ldots; x_m, y_m, r_m, v_m)$$

where

- m is the number of shocks in the group.
- x_i and y_i are the coordinates
- r_i is the radius (or time of formation)
- v_i is the speed
- θ_i is the direction

of each shock i in the sequence.

Graph Algorithms and Object Recognition: max clique

Measuring Shock Similarity

The similarity measure we use is a linear combination of four terms, incorporating the differences in lengths, radii, velocities, and curvature of two shock sequences.

Each term is normalized to provide a unitless quantity, so that these different geometric properties can be combined.

The measure provides a number between 0 and 1, which represents the overall similarity between the geometric attributes of the two nodes being compared.

The measure is designed to be invariant under rotations and translations of two shapes.

Graph Algorithms and Object Recognition: max clique

Results

Top 8 topological attributed matches								
Query	1	2	3	4	5	6	7	8
	0.000	0.257	0.344	0.455	0.470	0.471	0.481	0.507
	0.000	0.257	0.384	0.416	0.482	0.527	0.529	0.530
	0.000	0.344	0.384	0.447	0.498	0.516	0.520	0.529
	0.000	0.354	0.466	0.466	0.533	0.589	0.590	0.609
	0.000	0.314	0.459	0.466	0.477	0.563	0.567	0.574
	0.000	0.314	0.354	0.459	0.524	0.553	0.557	0.562
	0.000	0.194	0.302	0.354	0.375	0.445	0.537	0.560
	0.000	0.328	0.358	0.359	0.375	0.420	0.562	0.583
	0.000	0.194	0.309	0.310	0.358	0.378	0.507	0.532
	0.000	0.378	0.403	0.404	0.420	0.445	0.481	0.482
	0.000	0.281	0.309	0.354	0.359	0.403	0.536	0.539
	0.000	0.281	0.302	0.310	0.328	0.404	0.544	0.545
	0.000	0.114	0.471	0.471	0.524	0.527	0.558	0.560
	0.000	0.114	0.470	0.476	0.529	0.538	0.564	0.565
	0.000	0.471	0.476	0.480	0.486	0.519	0.539	0.562
	0.000	0.400	0.447	0.459	0.524	0.533	0.596	0.629
	0.000	0.400	0.459	0.466	0.477	0.530	0.550	0.568
	0.000	0.095	0.126	0.555	0.562	0.567	0.572	0.575
	0.000	0.095	0.160	0.544	0.563	0.587	0.588	0.613
	0.000	0.126	0.160	0.592	0.600	0.604	0.605	0.608
	0.000	0.334	0.366	0.416	0.455	0.498	0.518	0.519
	0.000	0.185	0.334	0.486	0.520	0.529	0.529	0.558
	0.000	0.185	0.366	0.480	0.527	0.529	0.531	0.560
	0.000	0.601	0.730	0.730	0.732	0.738	0.742	0.759
	0.000	0.601	0.680	0.681	0.689	0.692	0.692	0.693

Graph Algorithms and Object Recognition: max clique

Conclusions

A formal framework for matching hierarchical structures, non-hierarchical structures (free trees), attributed hierarchical structures, and noisy structures has been introduced.

the basic ingredients are:

- subtree isomorphism as clique search
- clique search as a quadratic program
- replicator equations as tree matching co-operative algorithms
  - -- easily implementable in H/W (Torsello and Pelillo, 1998)
  - -- offer the advantage of biological plausibility

Application to shape matching problem (via hierarchical shock trees and free shape-axis trees)

Extension to many-to-many matching (Pelillo et al, 2001).

Graph Algorithms and Object Recognition: max clique