Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Matching Random Tree Models of Spatio-Temporal Patterns to Tables or Graphs

David W. Paglieroni and Faranak Nekoogar
Lawrence Livermore National Laboratory
E-mail: {paglieroni1, nekoogar1}@llnl.gov

Abstract — The problem of matching random tree models of multi-component patterns to tables or graphs containing components extracted from diverse data sources is considered. We focus on bi-level trees whose branches emanate from one root node and terminate on different leaf nodes. Node and branch attributes are treated as random variables. Tree nodes represent pattern components of specified types that occur in tables or graphs to be searched. For each item in the table or graph with a type match to the tree root, there is a set of components from the table or graph that are candidate leaves for optimal matches to the tree model. We adopt a view of optimal matches to random tree models as minimum cost assignments of candidate leaves to tree branches. Model-based formulas are derived for computing costs associated with assignments of specific candidate leaf components from tables or graphs to specific tree branches.

We specify an ontology suitable for dynamic geo-spatial query problems in which (1) tree nodes represent physical objects or events on the ground (buildings, roads, communication transmissions…), and (2) branch attributes characterize, with uncertainty, distance or time separations between components, and angles between links connecting components. Our approach is used to search very large images for specific types of buildings in probabilistically constrained spatial arrangements, with the goal of ranking model matches for efficient inspection by human analysts.

Keywords: statistical graph, random tree model, bipartite graph, Hungarian algorithm, content-based image retrieval

I. Introduction

Relational data mining is concerned with finding or discovering multi-component patterns of interest in large information repositories that contain items previously extracted from various sources of structured or unstructured raw data (e.g., still overhead images, video, voice or email transactions over phone or computer networks, measurements from in-situ sensor networks over time, etc.). The underlying data structures for these information repositories were originally limited to tables. Each row in a table contains selected attributes for one item extracted from raw data. Commercial databases, search engines and Geographical Information Systems (GIS's) maintain tables containing items arranged a priori by type and attribute in a manner that allows items which satisfy prescribed type and attribute constraints to be rapidly retrieved.

Underlying data structures for information repositories have more recently evolved beyond tables to include graphs, which incorporate relationships between items. Graphs provide a topological representation of items extracted from raw data in which items are captured as vertices (nodes), and relationships between items are depicted as edges (links). Graph matching can be cast as a sub-graph isomorphism problem, in which the objective is to find topological matches to a sub-graph within a much larger graph. A survey of graph matching algorithms can be found in [17]. The discussion in [17] focuses on finding exact (complete) matches to sub-graphs, but some algorithms can also find inexact (incomplete) matches [19]. By way of very brief overview, three classes of commonly used methods for graph matching are structural matching, semantic matching, and graph feature matching. Structural methods consider only graph link structure, i.e., the vertices and edges are treated as un-typed and un-attributed. The basic method presented in [16] seeks all mappings of sub-graph vertices to graph vertices, but its complexity can grow exponentially with the number of nodes in the graph being searched. Greedy search methods based on depth-first search (low memory requirements) or breadth-first search (higher memory requirements) can also be expensive [5]. They can be hierarchical and combine patterns discovered on previous iterations [20]. Semantic methods consider not only graph link structure, but also types and attributes of vertices and edges ([6,18]). Graph feature matching methods match feature vectors constructed from graph invariants (the degree of each vertex, the number of sub-graph vertices or loops…) [9]. However, graph invariants are more typically used to improve search efficiency by reducing the search space through pre-screening and filtering. For example, the NAUTY algorithm uses graph invariants pre-computed for each vertex in the graph being searched as the basis for selecting candidate vertices [11]. All graph invariants can be derived from the graph adjacency matrix [8]. All row-wise and column-wise permutations of this matrix represent the same graph. The minimum description length (MDL) of a graph is the minimum number of bits needed to encode its adjacency matrix (taking into account sub-structure

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48, UCRL-CONF-225685.

redundancies in sub-structure hierarchies) [13]. The similarity between two graphs can be measured by encoding the difference between those graphs using the MDL principle (smaller MDL's imply greater similarities). Alternatively, the graph edit distance can be used [12]. This measure reflects the minimum cost of edit operations (insertions, deletions, or substitutions of vertices or edges) needed to transform one graph into the other.

Most prior work on relational pattern matching is based on matching relationally structured patterns (sub-graphs) to relationally structured databases (graphs). In contrast, our work focuses on matching patterns that are relationally structured in a spatio-temporal sense to relationally *unstructured* databases (tables). The structures of our relational patterns are modeled using trees (a special type of sub-graph). We argue that hybrid databases containing graphs, some of whose nodes are spatio-temporally anchored to tables, can be used to improve search precision and efficiency. Our emphasis is on tables maintained and indexed by a GIS populated with items manually or automatically extracted from overhead images.

A novel efficient approach to searching tables or graphs for a complete set of optimal exact or inexact matches to *random tree* (*RT*) *models* of multi-component patterns is developed in this paper. Random trees are random (statistical) graphs with one root node and at least one leaf node. Tree nodes are connected by directed links (branches). Branches cannot terminate on the root or emanate from leaves. However, trees can have internal nodes that branches both emanate from and terminate on. The attributes of RT nodes and branches are random variables. Although statistical graphs have been previously applied to search and data mining problems, we are introducing a novel method for quickly finding optimal matches to a useful class of statistical graph models in tables whose elements share a common spatio-temporal context.

RT model nodes represent components C_i that share a common spatio-temporal context, connected by branches or directed *links* L_k that specify probabilistically uncertain contextual relationships between components (see Fig.1). We focus on bi-level trees whose branches all emanate from the same root (a transmitter *T*) and terminate on different leaves (receivers R_i). Bi-level trees have no internal nodes. The ontology defines the types of components, their attributes, and the types of relationships supported. We specify an ontology suitable for dynamic geo-spatial query problems in which (1) tree nodes represent physical objects or events on the ground (buildings, vehicles, roads, communication transmissions / receptions, etc.), and (2) branch attributes characterize, with probabilistic uncertainty, distance or time separations between components, and angles between links connecting components.

Section II describes the structure of a bi-level RT model and how to characterize model uncertainty. We show that when long axis angles for root components are known, optimal model matches can be computed very efficiently because the model branches can be decoupled. A novel efficient algorithm for finding optimal exact and inexact matches to bi-level RT models in tables or graphs is developed in Section III. The algorithm operates on candidate roots and leaves retrieved from the table or graph using commercial search engine tools. These pre-screening tools improve search efficiency by reducing the search space. In Section IV, our approach is applied to content-based image retrieval problems that involve searching very large images for specific types of buildings in probabilistically constrained spatial arrangements, with the goal of ranking model matches from best to worst so that they can be efficiently inspected by human analysts.

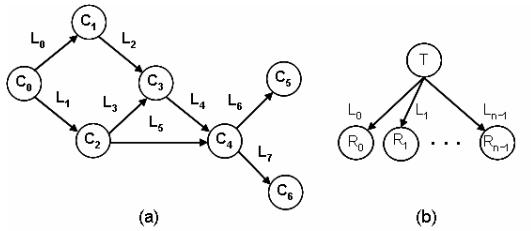


Fig.1 (a) Random graph model (b) bi-level random tree model.

II. Random Tree Models

A bi-level random tree (RT) model *Q* of a multi-component pattern that occurs in a table or graph to be searched is defined by (1) a set of components (one root node plus at least one leaf node) and their attributes, and (2) a set of branches (directed links from the root to leaves) and their attributes. All attributes can be treated as random variables.

A. Nodes and Links

Nodes representing components *C* associated with RT models *Q* have type, feature and context attributes (Table I). Feature attributes, which define component characteristics, will not be discussed in detail.

Let *P* be a pattern matched to model *Q*. The link L_k realized in pattern *P* emanates from component $C_{i_k}(P)$ in pattern *P* and terminates on component $C_{j_k}(P)$, where the indices i_k and j_k are specified in model *Q*. By requiring roots to be spatially localized, one can ensure that patterns *P* matched to models *Q* will be "anchored" to well-defined spatial locations. Leaves can either be spatially localized or spatially extended, and all components can be either transient or persistent.

TABLE I
Suggested Attributes for Component *C*

Attribute Type	Symbol	Description
Type	$\tau(C)$	Type ID for component *C*
	$s(C)$	Similarity ∈ [0,1] between *C* and model of type $\tau(C)$
Context	$\{[x(C), y(C), z(C)]\}$	Centroid for localized components *C*, polyline / polygon vertices for extended components *C* (length *m* sequence, $m = 1$ for localized components)
	$[t_0(C), t_1(C)]$	Time interval of observation or occurrence for *C*
	$\theta(C)$	Long axis angle for $C \in [-\pi,\pi)$

For a pair of components C_i and C_j, at least one of which is localized, there is a pair of associated *focus points* $[x(C_i,C_j), y(C_i,C_j), z(C_i,C_j)]$ and $[x(C_j,C_i), y(C_j,C_i), z(C_j,C_i)]$ for C_i and C_j respectively. If C_i is localized, the focus point is always the centroid. If C_i is extended, the focus point relative to some other localized component C_j is the point belonging to C_i closest to the centroid of C_j.

TABLE II
SUGGESTED ATTRIBUTES FOR LINK L_K

Symbol	Description	Remarks
d_k	Length of link	$d \geq 0$ and $\theta \in [-\pi,\pi)$ specify the length and pointing direction of L_k (θ is in the *xy* plane)
θ_k	Pointing direction of link ray	
t_{0k}, t_{1k}	Time delta of occurrence: leaf – root	0 if either component is persistent, computed for both start time (t_{0k}) and end time (t_{1k})
α_k, β_k	Angle from root / leaf long axis ray to directed link ray	α and $\beta \in [-\pi,\pi)$. For bi-level RT models, α can also provide information about angles between pairs of directed links

The values of all link attributes in Table II depend solely on the attributes of the linked root and leaf. Length d_k and pointing direction θ_k are computed from focus points for the pair $[C_{i_k},C_{j_k}] = [C_{i_k}(P),C_{j_k}(P)]$. t_{0k} and t_{1k} are differences between start and end times for those components. Also,

$$\alpha_k = \Delta(\theta(C_{i_k}), \theta_k), \quad \beta_k = \Delta(\theta(C_{j_k}), \theta_k) \qquad (1)$$

where $\Delta(\phi_j,\phi_k) \in [-\pi,\pi)$ is the angle from the ray pointing in direction $\phi_j \in [-\pi,\pi)$ to the ray pointing in direction $\phi_k \in [-\pi,\pi)$:

$$\Delta(\phi_j,\phi_k) = \phi_k - \phi_j + 2\pi n \quad n \in \{-1,0,1\}: \Delta(\phi_j,\phi_k) \in [-\pi,\pi) \qquad (2)$$

An expert can quantify the importance of component C_i to model Q by assigning a positive weight to C_i. The weights sum to one over all N components. To simplify the discussion, let us assume that all components C_i are equally important and thus have weights of $w(C_i) = 1/N$. If Q is a bi-level RT model, each of the $L = N-1$ links L_k will also be equally important and thus have weights $w(L_k) = 1/L$.

From equations (1)-(2), it can be shown that for a bi-level RT model,

$$\Delta(\alpha_j,\alpha_k) = \Delta(\theta_j,\theta_k) \qquad (3)$$

Thus, if the goal is to construct a bi-level RT model for which the angle from link L_j to link L_k is $\theta_{j,k}$, then α_j and α_k should be specified in the model such that $\Delta(\alpha_j,\alpha_k) = \theta_{j,k}$.

In bi-level RT models, when the admissible component types for the nodes have well-defined long axes, the values of all link attributes in Table II depend solely on the attributes of the linked root and leaf. This allows the links to be decoupled in the model matching process, and, as discussed in Section 3.2, bi-level RT model matching is much simpler when all branches can be decoupled.

B. Model Uncertainty

A graph model Q becomes random when the component and link attributes are treated as random variables. For example, an expert can impose probabilistic uncertainty on Q by treating link attributes $x = v_{k,i}$ $k = 0\ldots L-1$, $i = 0\ldots n_a-1$ as values of random variables X, and defining *relational constraint density functions* (*RCDF's*) $g_X(x)$ for those variables. RCDF's are scaled versions of PDF's that are non-negative and have a maximum value of unity. Examples of PDF's that are trivial to evaluate are *rect*(y) and *tri*(y), which for $y = (x-\mu)/\sigma$, represent rectangular and triangular pulses of width $\sigma > 0$ centered on $x = \mu$. If $\sigma = 0$, these functions become Kronecker delta functions $\delta(y)$.

The link attributes d, t_0, and t_1, and thus their RCDF's, are aperiodic. The domain of d is $x \geq 0$ and the domain of t is the set of all real numbers. The link attributes θ, α, and β, and thus their RCDF's, are periodic 2π. The fundamental period is $x \in [-\pi,\pi)$. For link L_k, the density functions and density function parameters imposed on each of its n_a attributes $v_{k,i}$ are captured in a relationships array $\boldsymbol{r_k}$ with n_a rows $\boldsymbol{r_{k,i}}$, $i = 0\ldots n_a-1$:

$$\boldsymbol{r_{k,i}} = [w(v_{k,i}), \text{density\_type}(v_{k,i}), \mu(v_{k,i}), \sigma(v_{k,i}), \Delta(v_{k,i})] \qquad (4)$$

where $w(v_{k,i})$ and $\Delta(v_{k,i})$ are the weight and period assigned to $v_{k,i}$. The weights are non-negative and sum to one. Link attributes not used by the model have weights of zero. The non-zero weights are typically equal.

All link attributes except β have unimodal RCDF's. As shown in Fig. 2, β is bimodal because the long axis computed for any elongated component can point in either of two opposite directions. The second *complementary mode* $\mu'(\beta_k) \in [-\pi,\pi)$ can be derived from the first mode $\mu(\beta_k) \in [-\pi,\pi)$ as

$$\mu'(\beta_k) = \mu(\beta_k) + (2n-1)\pi \quad n \in \{0,1\} : \mu'(\beta_k) \in [-\pi,\pi) \qquad (5)$$

Several domain experts have indicated that if an intuitive graphical interface for RT model specification were provided, they would be motivated to use it. We plan to develop a

simple interface for specifying node types and spatio-temporal relations between nodes, between branches or between branches and nodes. Policies tied to different levels of statistical uncertainty that assign default values of μ and σ to various RT model attributes will be provided. The expert will be able to override the defaults if desired.

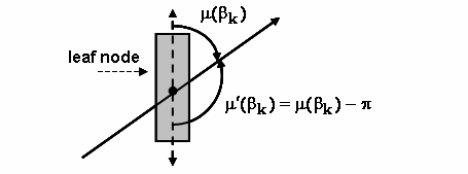


Fig.2 Geometry for leaf node complementary angles.

III. Matching Random Tree Models

A bi-level RT model can be specified as follows:

$$\begin{aligned} &Q : N && \text{(number of nodes, > 0)} \\ &\boldsymbol{\tau} = [\{\tau\}_0 \ldots \{\tau\}_{N-1}] && \text{(admissible node types)} \\ &\boldsymbol{r} = [\boldsymbol{r_0} \ldots \boldsymbol{r_{L-1}}] && \text{(probabilistic uncertainty)} \end{aligned} \quad (6)$$

Bi-level RT models are simpler than general random graph models. The number of links is $L = N–1$, and the indices for the nodes that link L_k emanates from and terminates on are

$$[i_k, j_k] = [0, k+1], \quad k = 0 \ldots L-1 \quad (7)$$

where component 0 is the root node, and component $k+1$ is the leaf node attached to link k. Also, we have assumed for simplicity that all component weights are equal. Examples of bi-level RT models are given in Section IV.

Let Q be a bi-level RT model for a multi-component spatio-temporal pattern. An efficient approach to retrieving all optimal exact or inexact pattern matches P to model Q from a table is developed in this Section and can be summarized as follows:

1. Retrieve all candidate components from the table whose types and attributes satisfy constraints imposed by Q on the root node.
2. For each candidate root node, retrieve all candidate leaf nodes that satisfy constraints imposed by Q on leaf node type, feature, and context attributes.
3. For each candidate root, find the optimal match to Q from among all of its candidate leaves, and compute the degree of match.

The first two steps leverage commercial search engine tools. The last step uses a customized code layer based on algorithms developed in the remainder of this Section.

Our RT models explicitly capture spatio-temporal relations between nodes. One could likewise choose to explicitly capture spatio-temporal relations between spatio-temporally proximate items extracted from raw data sources in graphs rather than tables. Step 2 is actually simpler for graphs than for tables because candidate leaves are limited to graph nodes connected to candidate root nodes through some level of indirection. However, spatio-temporal relations are handled far more efficiently by tables because they store spatio-temporal attributes for each item rather than spatio-temporal relations between each pair of items. Graphs, on the other hand, are better for representing logical relations between nodes. A table-graph hybrid can be created by flagging certain graph nodes as spatio-temporally anchored. Although each anchored node points to a specific item in the table, not all table items need to have corresponding graph nodes. The advantage of hybrid databases is that in steps 1-2, candidate roots and leaves from the table can be restricted not just by type and spatio-temporal context, but also by logical relations to graph nodes.

A. Model Evaluation

Consider a pattern $P = \{C_0(P) \ldots C_{N-1}(P)\}$ characterized by a set of $N > 1$ retrieved components. Our immediate goal is to compute the degree of match similarity $S(P;Q) \in [0,1]$ between pattern P and bi-level RT model Q (1 for perfect similarity). Like Q, P has N components, where component C_i in P corresponds to component C_i in Q. If component C_i is missing from P, then $s(C_i(P)) = 0$. Assume $\tau(C_i(P)) \in \{\tau\}_i$.

$S(P;Q)$ can be viewed as a function of P. A *model evaluation* for model Q is a calculation of $S(P;Q)$ for some specific pattern P. A model evaluation can be expressed as the sum of *link evaluations* $S_k(P;Q) \geq 0$ for each of $L > 0$ links L_k, $k = 0 \ldots L-1$:

$$S(P;Q) = \sum_{k=0}^{L-1} S_k(P;Q) \in [0,1] \quad (8)$$

If all components have equal weights, $S(P;Q)$ can be intuitively expressed as the mean of the root component s value and penalized leaf component s values, where the penalty factors account for relational inconsistencies between links in P and corresponding links in Q:

$$S(P;Q) = \frac{1}{N}\left[s(C_0(P)) + \sum_{k=0}^{L-1} g_k(P;Q) \cdot s(C_{k+1}(P)) \right] \quad (9)$$

If Q contains a root but no leaves, then $N = 1$, and from equation (9), $S(P;Q)$ is just the s value of the root component. Combining equations (8)-(9) yields a simple expression for link evaluation:

$$S_k(P;Q) = \frac{1}{N}\left[\frac{s(C_0(P))}{L} + g_k(P;Q) \cdot s(C_{k+1}(P)) \right] \in [0,1] \quad (10)$$

In equations (9)-(10), the relational consistency factor $g_k(P;Q) \in [0,1]$ quantifies the relational consistency between links L_k

in P and Q. $g_k(P;Q)$ is computed as a weighted sum of relational consistencies associated with the link attributes when all attributes used in the model meet the basic constraints on their values:

$$g_k(P;Q) = \left[\prod_{i=0}^{n_a-1} \tilde{g}_{k,i}(\, v_{k,i}(P)\,;\,Q)\right] \cdot \left[\sum_{i=0}^{n_a-1} w(v_{k,i}) \cdot g_{k,i}(\, v_{k,i}(P)\,;\,Q)\right] \quad (11)$$

$g_{k,i}(x;Q)$ is the RCDF for $v_{k,i}$. $\tilde{g}_{k,i}(x;Q) = 0$ if $w(v_{k,i}) > 0$ and $g_{k,i}(x;Q) = 0$. $\tilde{g}_{k,i}(x;Q) = 1$ otherwise.

B. Bipartite Graph Matching

For a given candidate root, any bi-level RT model Q with $m > 0$ candidate leaves can be transformed into a bipartite graph in which bipartite graph nodes of one "color" correspond to m distinct candidate leaves R_i, $i = 0 \ldots m-1$ for RT model links, and bipartite graph nodes of the other "color" correspond to the $L > 0$ distinct RT model links L_j, $j = 0 \ldots L-1$ (Fig.3). All bipartite graph nodes representing candidate leaves are placed into one column. All bipartite graph nodes representing RT model links are placed into a second column. Bipartite graph nodes in the same column cannot be connected. R_i is a candidate leaf for L_j if and only if there is an edge connecting node R_i to node L_j in the bipartite graph. The edge connecting R_i to L_j is assigned a cost $c_{i,j} = 1 - S_{i,j} \in [0,1]$, where

$$S_{i,j} \triangleq \begin{cases} S_j(P;Q),\ P : C_j(P) = R_i & \text{edge connects } R_i \text{ to } L_j \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The 2D matrix of cost values can be transformed into a binary solution matrix in which the elements of value 1 assign candidate leaves to links so as to minimize cost (maximize similarity to Q). For bipartite graphs with n nodes (the number of candidate leaves plus RT model links), basic implementations of the Hungarian algorithm find a minimal cost assignment in worst case $O(n^3)$ time [1,10].

To allow for the possibility that optimal pattern matches can be inexact (i.e., that certain leaves can be missing), the cost matrices can be augmented with $L-1$ additional rows corresponding to $L-1$ *phantom leaves* R' (i.e., missing as opposed to real leaves). Phantoms can be attached to no more than one RT model link at a time, but they are "floating", i.e., each is a candidate for all RT model links and can thus be attached to any RT model link. There should be exactly $L-1$ phantoms in the augmented bipartite graph because (a) there can be no more than L phantoms when there are only L links, (b) the solution characterized by assigning a phantom to each of the L links is sub-optimal when $m > 0$, and (c) the optimal solution drawn from a solution space that contains $L-1$ phantoms will be at least as good as the optimal solution drawn from a solution space containing fewer than $L-1$ phantoms. If a phantom leaf is attached to RT model link L_k, then $s(C_{k+1}) = 0$ in equation (10), and the link L_k evaluation simplifies to

$$S_k(P;Q) = s(C_0(P)) / (NL) \quad (13)$$

When the RT model components have well-defined long axes, the values of variables for one link do not depend on values of variables for other links. In this case, for a given candidate root, only 2 minimal cost assignment problems will need to be solved in order to find the optimal pattern match to a bi-level RT model Q because its links are decoupled. Model Q and the *complementary model* Q' will both need to be matched. Q' differs from Q only in that the root long axis is reversed. This means that for each link L_k, $\mu(\alpha_k)$ in Q is replaced by $\mu'(\alpha_k)$ in Q' (see equation (5) with β replaced by α). If the model component long axes are not well-defined, then for a given candidate root, $2K$ minimal cost assignment problems will need to be solved in order to find the optimal pattern match to a bi-level RT model Q, where K is the number of candidate leaves for link L_0 (the *reference link*).

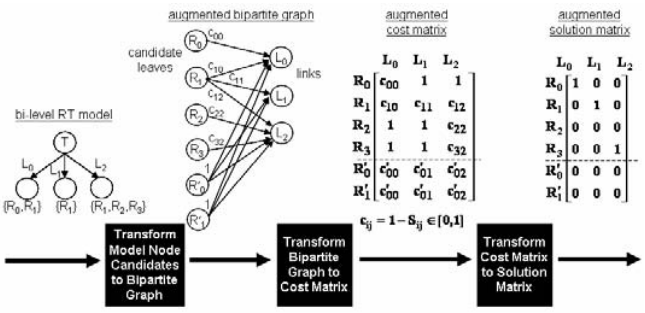


Fig.3 Illustration of bipartite graph matching approach to bi-level RT model matching.

IV. Large Image Search and Content-Based Image Retrieval Examples

When one considers that large overhead images are typically searched block-by-block, it becomes clear that the large image search problem is closely related to content-based image retrieval (CBIR) problems involving large numbers of small images. Most algorithms and systems for CBIR attempt to derive compact sets of features that provide incomplete characterizations of small images, and then match those features to features characteristic of desired images. Image features for CBIR are normally based on color and brightness [15], texture [7], edges [21], shapes of regions [4], etc. CBIR surveys can be found in [2,14].

Feature-based CBIR algorithms often produce search results that resemble desired images in some general way. On the other hand, graph-based CBIR algorithms are better suited for finding areas in large images that contain prescribed types of objects in certain spatial configurations. For example, in [3], entities extracted from images are represented on attributed relational graphs (ARG's) and matched to ARG

models that capture desired spatial configurations. Unlike feature-based methods, graph-based methods are relatively insensitive to scene background clutter. The inputs to graph-based methods are not images, but rather tables or graphs containing items extracted from images. These tables and graphs can just as easily contain items extracted not just from images, but also from multiple diverse data sources. Graph-based methods thus easily generalize beyond images to retrieval of patterns whose components represent items extracted from multiple data sources. Furthermore, the graph ontology can be readily altered to handle patterns in other than a spatio-temporal context

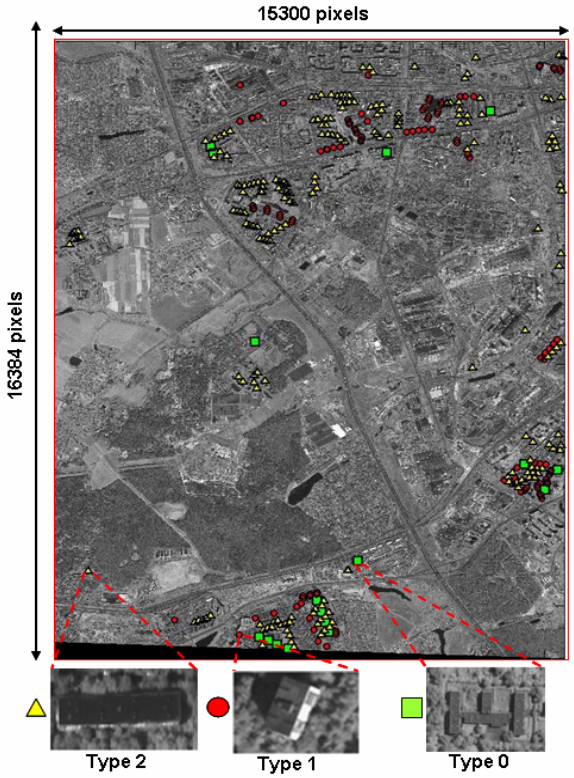


Fig.4 Building distributions by type (courtesy of DigitalGlobe).

In this Section, bi-level RT model matching is used to search for spatially constrained configurations of specific types of buildings in large overhead images. Relative to search strategies for individual buildings, RT model matching for specific building configurations has inherent false alarm mitigation properties based on spatial context (and more generally, spatio-temporal context). A table was generated from a large image of size 16384x15300 pixels (see Fig.4). Each row contains four attributes for one building: (1) building type, (2) building pixel coordinates, (3) building orientation (in image space), and (4) the degree of similarity between the building and its physical model (a measure of quality). Since our current objective is to demonstrate the bi-level RT model matching approach, it is not particularly important where the table comes from. Although the table was generated manually, it could have been generated with buildings extracted from the image by a computer, except that, one would expect the detection rate to be higher and the false alarm rate to be lower for buildings extracted manually. Fig.4 shows the spatial distribution for three types of buildings of interest in the subject image (courtesy of DigitalGlobe).

Two bi-level RT models were constructed. The first model represents a pattern with relatively constrained spatial relationships between buildings. The second model is less constrained.

A. Experimental Results for Model 1

In model 1, a specific spatial arrangement of type 0 and type 2 buildings is considered. Fig.5 depicts the physical geometry for model 1. Fig.6 depicts the bi-level random tree for model 1 graphically. The underlying PDF's for all link attributes are assumed to be rectangular pulse functions.

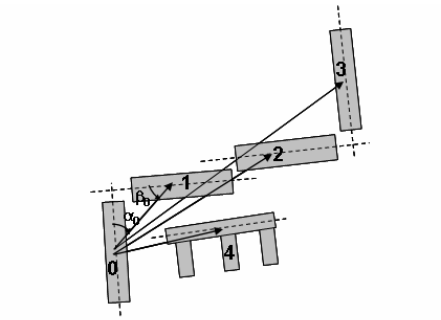
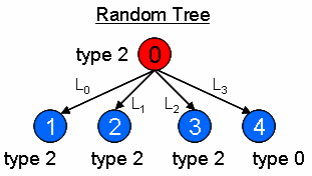


Fig.5 Physical geometry for bi-level RT model 1.



Uncertainty Parameters for Link Attributes

	k	μ	σ
d_k	0	120	40
	1	245	60
	2	360	60
	3	160	60
α_k, β_k	0	40, −50	20, 20
	1	60, −30	20, 20
	2	50, 50	20, 20
	3	85, 0	20, 20

Fig.6 Bi-level RT model 1.

Fig.7 shows thumbnail images corresponding the three best unambiguous RT model matches in order, and then three lesser matches selected arbitrarily for comparison. The first three thumbnails correspond to the only occurrences of this building configuration in the image, and they are exact (complete) matches. The remaining four thumbnails correspond to inexact (incomplete) matches. Fig.8 shows a plot of *S* value versus rank. The *S* values are so much higher for true matches than for false matches because the building configuration represented by model 1 is so highly constrained.

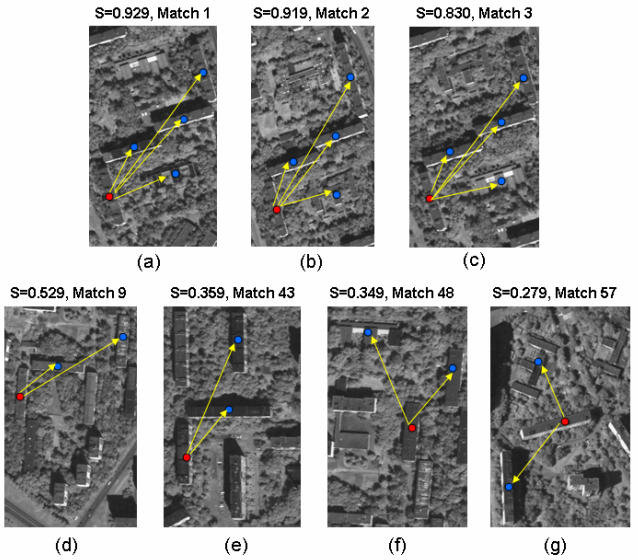


Fig.7 Matching results for bi-level RT model 1. (a)-(c) Top 3 matches. (d)-(g) 4 randomly chosen matches with lower ranks.

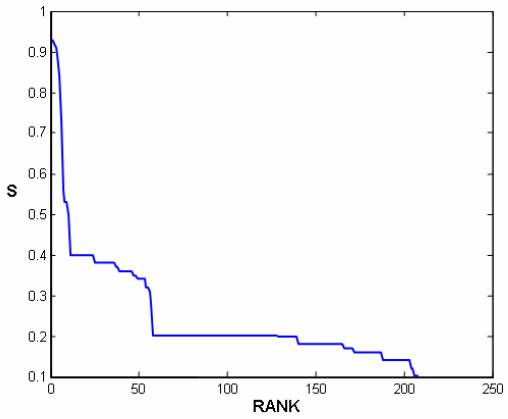


Fig.8 S value versus rank for matches to bi-level RT model 1.

B. Experimental Results for Model 2

In model 2, a specific roughly linear configuration of type 1 buildings is considered. The root corresponds to a building at one end of the sequence. Fig.9 depicts the physical geometry for model 2. Fig. 10 depicts the bi-level random tree for model 2 graphically. The underlying PDF's for all link attributes are assumed to be rectangular pulse functions. Model 2 constrains the linearity of building clusters to some degree, but allows flexibility in relative angles of buildings. The intent is to capture clusters of more-or-less linearly arranged type 1 buildings at various orientations.

Fig.11 shows thumbnail images corresponding the five best unambiguous RT model matches in order, and then three lesser matches selected arbitrarily for comparison. Fig.12 shows a plot of *S* value versus rank. For model 2, the S values do not decrease as abruptly from true to false matches as for model 1 because the building configuration represented by model 2 is less highly constrained.

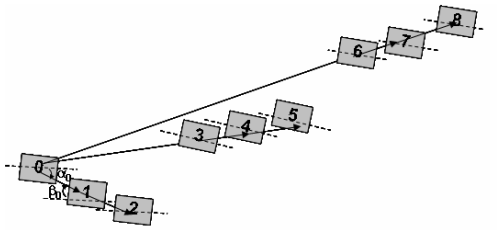
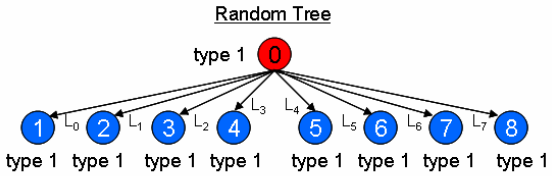


Fig.9 Physical geometry for bi-level RT model 2.



Uncertainty Parameters for Link Attributes

	k	μ	σ
d_k	0	50	30
	1	100	50
	2	235	90
	3	285	30
	4	335	50
	5	480	100
	6	530	60
	7	580	60
α_k, β_k	0-7	0, 0	30, 60

Fig.10 Bi-level RT model 2.

The results shown for bi-level RT models 1 and 2 are typical of results obtained with a variety of models. Since optimal RT model matches can be exact or inexact, highly ranked matches tend to be complete, and lower ranked matches tend to be increasingly incomplete.

V. CONCLUSIONS

A novel efficient approach for finding a complete set of optimal exact or inexact matches to bi-level random tree (RT) models of spatio-temporal patterns in tables has been developed. Our approach uses customized code layered on commercial search engine code, and jointly accounts for (1) the relative importance of each component to the model, (2) the similarity between corresponding components in the pattern match and model, (3) the relational consistency between the pattern match and model, (4) probabilistic uncertainty in the model, and (5) links missing from the pattern match. Our approach was successfully demonstrated by searching large overhead images for loosely and more tightly constrained building configurations. Topics for future study include elimination of redundant matches, comparison of matching performance with and without relational constraints, and realization of multi-level RT models with internal nodes as hierarchically nested bi-level RT models.

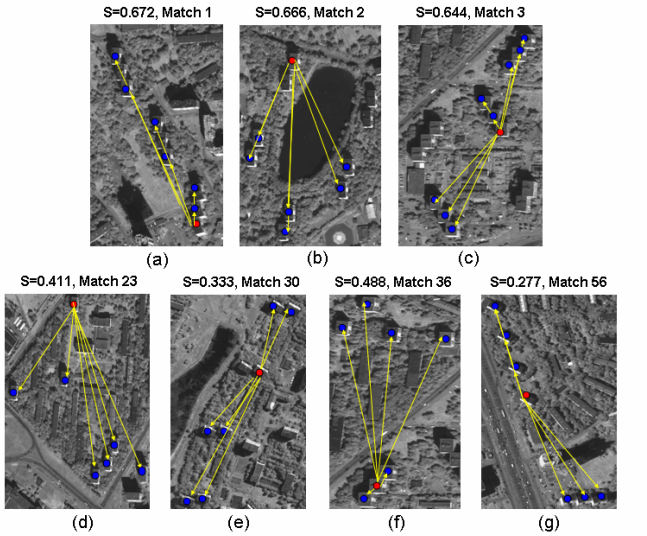


Fig.11 Matching results for bi-level RT model 2. (a)-(c) Top 3 unambiguous matches. (d)-(g) 2 randomly chosen matches with lower ranks.

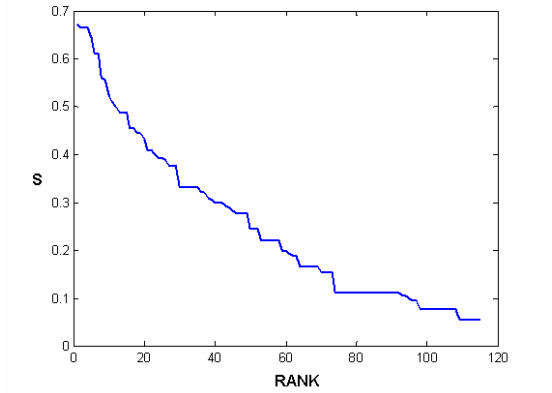


Fig.12 S value versus rank for matches to bi-level RT model 2.

ACKNOWLEDGEMENTS

The authors wish to thank Scott Kohn for suggesting that our random tree model matching problem might be addressed using bipartite graph matching techniques. Thanks also to Tom Edmunds and Carol Meyers for providing insight into the Hungarian algorithm.

REFERENCES

[1] R. Ahuja, T. Magnanti, J. Orlin, "Network Flows", Prentice Hall, 1993, pp.122, 471-472.

[2] S. Antani, R. Kasturi, R. Jain, "A Survey on the use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video", Pattern Recognition, Vol.35, 2002, pp.945-965.

[3] S. Berretti, A. Del Bimbo, E. Vicario, "Efficient Matching and Indexing of Graph Models in Content-Based Retrieval", IEEE Trans. Pattern Anal. Mach. Intell., Vol.23, No.10, October 2001, pp.1089-1105.

[4] Y. Chen, J. Wang, "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval", IEEE Trans. Pattern Anal. Mach. Intell., Vol.24, No.9, September 2002, pp.1252-1267.

[5] J. Cook, L. Holder, "Substructure Discovery Using Minimum Description Length and Background Knowledge", Journal of Artificial Intelligence Research, Vol.1, 1994, pp.231-255.

[6] R. Giugno, D. Shasha, "Graphgrep: A Fast and Universal Method for Querying Graphs", Proc. IEEE Int. Conf. Pattern Recognition, 2002, pp.112-115.

[7] P. Huang, S. Dai, "Image Retrieval by Texture Similarity", Pattern Recognition, Vol.36, 2003, pp.665-679.

[8] A. Inokuchi, T. Washio, H. Motoda, "Complete Mining of Frequent Patterns from Graphs: Mining Graph Data", Machine Learning, Vol.50, 2003, pp.321-354.

[9] H. Kashima, A. Inokuchi, "Kernels for Graph Classification", Proc. Int. Workshop on Active Mining", 2002, pp.31-35.

[10] H. Kuhn, "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, 2, 1955, pp.83-97.

[11] B. McKay, NAUTY Users Guide (v1.5), Tech. Report TR-CS-90-02, Dept. of Comp. Sci., Australian National University, 1990.

[12] R. Meyers, R. Wilson, E. Hancock, "Bayesian Graph Edit Distance", IEEE Trans. Pattern Anal. Mach. Intell., Vol.22, No.6, June 2000, pp.628-635.

[13] J. Rissanen, **Stochastic Complexity in Statistical Inquiry**, World Scientific Publishing Company, 1989.

[14] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-Based Image Retrieval at the End of the Early Years", IEEE Trans. Pattern Anal. Mach. Intell., Vol.22, No.12, December 2000, pp.1349-1380.

[15] M. Swain, B. Ballard, "Color Indexing", Int. J. Computer Vision, Vol.7, No.1, 1991, pp.11-32.

[16] J. Ullman, "An Algorithm for Subgraph Isomorphism", J. ACM, Vol.23, No.1, 1976, pp.31-42.

[17] T. Washio, H. Motoda, "State of the Art in Graph-Based Data Mining", ACM SIGKDD Explorations Special Issue on Multi Relational Data Mining, Vol.5, No.1, 2003, pp.59-68.

[18] M. Wolverton et al, "LAW: A Workbench for Approximate Pattern Matching in Relational Data", Proc. Innovative Applications of Artificial Intelligence Conf., 2003, pp.143-150.

[19] M. Wolverton, I. Harrison, J. Lowrance, A. Rodriguez, J. Thomere, "Software Supported Pattern Development in Intelligence Analysis", Pub. 1147, AI Center, SRI International, 2006.

[20] K. Yoshida, H. Motoda, N. Indurkhya, "Graph-Based Induction as a Unified Learning Framework", J. of Applied Intel., Vol.4, 1994, pp.297-328.

[21] X. Zhou, T. Huang, "Edge-Based Structural Features for Content-Based Image Retrieval", Pattern Recognition Letters, Vol.22, 2001, pp.457-468.