1 Background

When a person interested in a topic enters a keyword into a Web search engine, the response is nearly instantaneous (and sometimes overwhelming). The impressive speed is due to clever inverted index structures, caching, and a domain-independent knowledge of strings. Our project seeks to construct algorithms, data structures, and software that approach the speed of keyword-based search engines for queries on structural databases.

A structural database is one whose data objects include trees, graphs, or a set of interrelated labeled points in two, three, or higher dimensional space. Examples include databases holding (i) protein secondary and tertiary structure, (ii) phylogenetic trees, (iii) neuroanatomical networks, (iv) parse trees, (v) molecular diagrams, and (vi) XML documents. Comparison queries on such databases require solving variants of the graph isomorphism or subisomorphism problems (for which all known algorithms are exponential), so we have explored a large heuristic space.

2 Prototype

We have two different search techniques: one based on topological relationships and the other based on physical position. In SIGMOD 2000 we will demo a search engine capable of answering the following queries on a structural database $D$ containing 3D graphs:

1. [nearest-neighbor query] Given a query object $q$, which items $d'$ in $D$ are “closest” to $q$? We have experimented with two different similarity scores. The score between a graph $g_1$ and a graph $g_2$ based on an alignment $M$ is defined as the fraction of edges in $g_1$ ($g_2$, respectively) that are consistent in $g_2$ ($g_1$, respectively) based on $M$. Edge $(i, j)$ of $g$ has a consistent edge in $g'$ based on the alignment $M$ if $M[i][j] = 1$. The larger the score between $g_1$ and $g_2$, the closer $g_1$ is to $g_2$. We use this query to illustrate our search technique based on topological relationships.

2. [discovery query] Which substructures approximately occur in all items in $D$? A substructure may occur in a graph after allowing for an arbitrary number of whole-structure rotations and translations as well as a small number (specified by the user) of edit operations in the substructure or in the graph. (When a substructure appears in a graph only after the graph has been modified, we call that appearance “approximate occurrence.”) The edit operations include relabeling a node, deleting a node and inserting a node. We use this query to illustrate our discovery technique based on physical position. We can search by physical position and discover by topological relationships.

For both queries, we will show the best alignment information in addition to graphically displaying the qualifying (sub)structures.