Research in algorithms for geometric pattern matching

Piotr Indyk

September 7, 2001

1 Project overview

The goal of this project is to develop efficient algorithms for a variety of geometric pattern matching (GPM) problems. A typical geometric pattern matching problem involves a *pattern* P and a *scene* S (or possibly a database of scenes $S_1 \dots S_n$), and the goal is to discover one or all occurrences of P in the scene(s). Geometric pattern matching is pervasive in many areas of computer science. It is of special importance in *computer vision*, where it directly corresponds to fundamental vision problems like object registration and recognition. It is also of importance in *computational drug design* and *computational biology*, where it has been successfully used for identification of drug molecules with similar shapes (and therefore similar chemical properties). More recent motivation comes from the field of *visual information retrieval.* In this case we are given a database of images and a pattern, and the goal is to find an image containing the pattern. So far, most of the visual search engines compute the similarity between the pattern and the image using only global features (e.g., color histogram, texture characteristics etc). However, in order to improve the search quality, using the geometry of the image and the pattern seems unavoidable.

2 Matching with one scene

In the simplest version of a GPM problem we are given a pattern and only one scene. Typically, both the pattern and a scene are represented by a set (or an ordered sequence) of features. In the simplest case, each feature is just a point; in general, more elaborate features (e.g., edges or splines) can be allowed. In addition, we are given a *distance function* $D(\cdot, \cdot)$, which for any two objects, represented by feature sets S, S', specifies the dis-similarity $D(S, S')$ between S and S'. Once the distance function is defined, the problem can be formally defined as follows: given a pattern P and scene S, find a subset $S' \subset S$ of a scene such that $D(P, S')$ is minimized. Alternative formulations of this problem involve finding one or all S''s such that $D(P, S')$ is smaller than predefined threshold etc.

Several distance functions $D(\cdot, \cdot)$ have been proposed in the literature. One of the most popular measures is the *Hausdorff distance*, defined as

$$D_H(S, S') = \max_{p \in S} \min_{p' \in S'} ||p - p'||_2$$

Several variants of this measure exist, including *symmetric* or *translation/rotation-invariant* Hausdorff distance. One can define further variations of this distance, e.g., by replacing the "max" in the above definition by "sum", etc. The Hausdorff distance has been introduced to computer vision and computational geometry by Huttenlocher and has been widely used since then.

Another popular measure, defined for *ordered* sequences of features, is the *Frechet distance*, closely related to the *time-warping distance.* Given two sequences of features (e.g., points) $S = s_1 \dots s_n$ and $S' = s'_1 \dots s'_m$, the Frechet distance $D_F(S, S')$ is defined as

$$D_F(S, S') = \min_{\pi} \max_{i=1 \dots n} ||s_i - s'_{\pi(i)}||_2$$

where π ranges over all monotone mappings from $\{1 \dots n\}$ to $\{1 \dots m\}$. Intuitively, this distance function measures the maximum distance between "corresponding" features of S and S', with respect to the best "correspondence" function π. As before, one can make the measure translation/rotation invariant, or replace "max" by "sum" (in the latter case the measure is called *time-warping distance*). The Frechet or time-warping distances were introduced in the speech-recognition community and are used e.g., for measuring distance between electronic pen signatures.

2.1 Previous and proposed work

Several efficient algorithms for solving pattern matching problems are known. For the translation-invariant Hausdorff distance, the pattern matching problem can be solved (approximately) in $O(n \log n)$ time[1] [Schulman'99,Indyk'99]; exact solutions can be found in polynomial time. The Frechet or time-warping distance are computationally more difficult: the best algorithm just for computing the distance $D_F(S, S')$ (even approximately) takes $O(n^2)$ time. The quadratic running time is a severe bottleneck which prohibits using these measures for large data sets. Only very recently a polynomial time algorithm for computing translation-invariant Frechet distance has been discovered [Efrat, Indyk, Venkatasubramanian'01].

Our goal is to design more efficient algorithms for solving (approximate) pattern matching problems under Frechet distance and/or similar measures. Once such algorithms are developed, we plan to perform experimental evaluation of their behavior and apply them to application areas where they can be used best.

3 Matching with many scenes

In this scenario we are given a pattern P and database consisting of *many* sets $S_1 \dots S_n$. Our goal is to solve the *nearest neighbor problem*, i.e., find S_i which minimizes $D(P, S_i)$, for a prespecified distance function $D(\cdot, \cdot)$ (as before, we could replace $D(P, S_i)$ by $D(P, S_i')$ for $S_i' \subset S_i$). A naive way of solving this problem is to compute $D(P, S_i)$ for all $i = 1 \dots n$. This approach, however, inevitably requires at least $\Omega(n)$ running time per pattern, which in most situation, is too expansive. Ideally, one would like to build an indexing structure which would identify the S_i minimizing $D(P, S_i)$ by inspecting only very few S_i's.

3.1 Previous and proposed work

Until recently, no sublinear (i.e., $o(n)$-time) algorithm for the nearest neighbor problem under *any* of the above measure was known. Only very recently it was shown [Farach-Colton,Indyk'99] how to construct a data structure which enables to find approximate nearest neighbor under Hausdorff distance in $\log^{O(1)} n$ time per pattern. So far, the data structure requires a considerable amount of memory, at least in theory. It is conceivable that it can be made to require much less storage in practice. No similar algorithms for other measures are known.

Our goal is to design a practical version of the data structure of [Farach-Colton,Indyk] and apply it to registration and recognition problems. In addition, we plan to develop efficient data structures for other measures, like Frechet and/or time-warping distance.

4 The Budget

The budget is roughly broken down as follows: 1 RA (55K), 1 month summer salary for PI (25K), Travel (10K), Equipment (10K). Total: 100K.

[1] n denotes the size of the input.