Point Matching Under Affine Transformation

John E. Hopcroft and Daniel P. Huttenlocher

Computer Science Department, Cornell University, 4130 Upson Hall, Ithaca NY 14853

Abstract

An important geometric problem in pattern matching and robotics is the determination of whether there exists an affine transformation (a general linear transformation and a translation) that maps each point in a set A onto a corresponding point in a set B. Restricted versions of this problem, where the transformation is an isometry or similarity, and where $|A| = |B|$ have been the subject of recent investigations. In this paper we consider the general problem of determining whether there is a two-dimensional affine transformation, $T : A \rightarrow B$, where $|A| \leq |B|$. That is, the set B is allowed to contain extraneous points, but each point of A must have a matching point in B.

In the case of matched cardinality point sets, we have developed an optimal $O(n \log n)$ algorithm for determining the existence of a transformation, T. The method relies on the affine properties that the centroid of a point set and the area ratios of triangles are preserved. The basic idea is to compute for each set A and B the areas of the triangles formed by connecting the centroid to each point in the set. Ratios of neighboring triangle areas are computed for the two point sets, and an affine transformation exists exactly when the ordered ratios are the same for the two sets.

If $|A| < |B|$ then there can be $O(n^3)$ transformations from A to B. For example this occurs when the points of A form a square and the points of B are on a regularly spaced grid. In general the number of transformations will be much smaller, so we have developed an *output sensitive* algorithm that runs in time $O(n^2 \log n + tm \log n)$, where $m = |A|$, $n = |B|$, and t is the number of transformations exhibited. The method uses the affine properties that intersection points and length ratios along a line are preserved.

The algorithm has three basic steps. In the first step, four points in A are chosen and the intersection is computed for the two crossing edges that connect the four points. The relative position of this intersection point along the two edges is preserved under an affine transformation. In the next step, the edge corresponding to each pair of points in B is marked using the relative positions from Step 1. Two marks will coincide only for a set of four points in B that are an affine transformation of the four points in A. The marking operation and a subsequent sorting require $O(n^2 \log n)$ time. Finally, each of the matches from Step 2 is checked by transforming every point of A and performing a range query to find a matching point of B.