Hausdorff Distance between convex polygons

1. Introduction

When talking about distances, we usually mean the shortest: for instance, if a point X is said to be at distance D of a polygon P, we generally assume that D is the distance from X to the nearest point of P. The same logic applies for polygons: if two polygons A and B are at some distance from each other, we commonly understand that distance as the shortest one between any point of A and any point of B. Formally, this is called a *minimin* function, because the distance D between A and B is given by :

$$D(A,B) = \min_{a \in A} \{ \min_{b \in B} \{ d(a,b) \} \} \qquad \text{(eq. 1)}$$

This equation reads like a computer program: « *for every point a of A, find its smallest distance to any point b of B ; finally, keep the smallest distance found among all points a* ».

That definition of distance between polygons can become quite unsatisfactory for some applications; let's see for example fig. 1. We could say the triangles are close to each other considering their shortest distance, shown by their red vertices. However, we would naturally expect that a small distance between these polygons means that no point of one polygon is far from the other polygon. In this sense, the two polygons shown in fig. 1 are not so close, as their furthest points, shown in blue, could actually be very far away from the other polygon. Clearly, the shortest distance is totally independent of each polygonal shape.

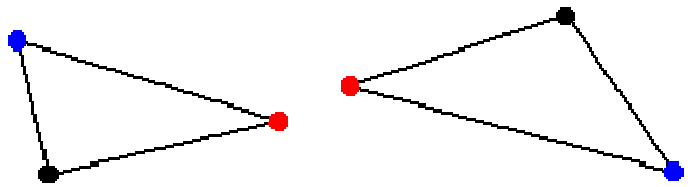

Figure 1 : The shortest distance doesn't consider the whole shape.

Another example is given by fig. 2, where we have the same two triangles at the same shortest distance than in fig. 1, but in different position. It's quite obvious that the shortest distance concept carries very low informative content, as the distance value did not change from the previous case, while *something* did change with the objects.

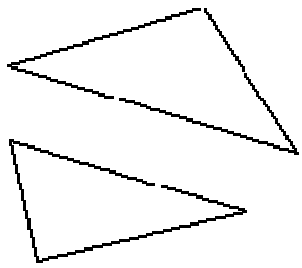

Figure 2 : The shortest distance doesn't account for the position of the objects.

As we'll see in the next section, in spite of its apparent complexity, the Hausdorff distance does capture these subtleties, ignored by the shortest distance.

2. What is Hausdorff distance ?

Named after Felix Hausdorff (1868-1942), Hausdorff distance is the « *maximum distance of a set to the nearest point in the other set* » [Rote91]. More formally, Hausdorff distance from set A to set B is a *maximin* function, defined as

$$h(A, B) = \max_{a \in A} \{ \min_{b \in B} \{ d(a, b) \} \} \qquad \text{(eq. 2)}$$

where a and b are points of sets A and B respectively, and d(a, b) is any metric between these points ; for simplicity, we'll take d(a, b) as the Euclidian distance between a and b. If for instance A and B are two sets of points, a brute force algorithm would be :

Brute force algorithm :

1. h = 0
2. for every point a_i of A,
   2.1 shortest = Inf ;
   2.2 for every point b_j of B
   $d_{ij} = d(a_i, b_j)$
   if d_{ij} < shortest then
   shortest = d_{ij}
   2.3 if shortest > h then
   h = shortest

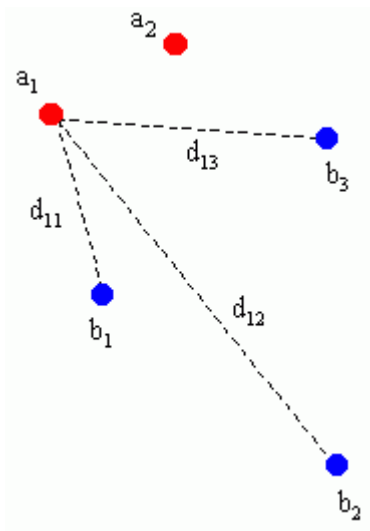

Figure 3 : Hausdorff distance on point sets.

This algorithm obviously runs in O(n m) time, with n and m the number of points in each set.

It should be noted that Hausdorff distance is *oriented* (we could say *asymmetric* as well), which means that most of times h(A, B) is not equal to h(B, A). This general condition also holds for the example of fig. 3, as h(A, B) = d(a1, b1), while h(B, A) = d(b2, a1). This asymmetry is a property of maximin functions, while minimin functions are symmetric.

A more general definition of Hausdorff distance would be :

$$H(A, B) = \max \{ h(A, B), h(B, A) \} \qquad \text{(eq. 3)}$$

which defines the Hausdorff distance ***between*** A and B, while eq. 2 applied to Hausdorff distance ***from*** A ***to*** B (also called *directed* Hausdorff distance). The two distances h(A, B) and h(B, A) are sometimes termed as *forward* and *backward* Hausdorff distances of A to B. Although the terminology is not stable yet among authors, eq. 3 is usually meant when talking about Hausdorff distance. Unless otherwise mentionned, from now on we will also refer to eq. 3 when saying "Hausdorff distance".

If sets A and B are made of lines or polygons instead of single points, then H(A, B) applies to all defining points of these lines or polygons, and not only to their vertices. The brute force algorithm could no longer be used for computing Hausdorff distance between such sets, as they involve an infinite number of points.

So, what about the polygons of fig. 1. Remember, some of their points were close, but not all of them. Hausdorff distance gives an interesting measure of their mutual proximity, by indicating the maximal distance between *any* point of one polygon to the other polygon. Better than the shortest distance, which applied only to *one* point of each polygon, irrespective of all other points of the polygons.

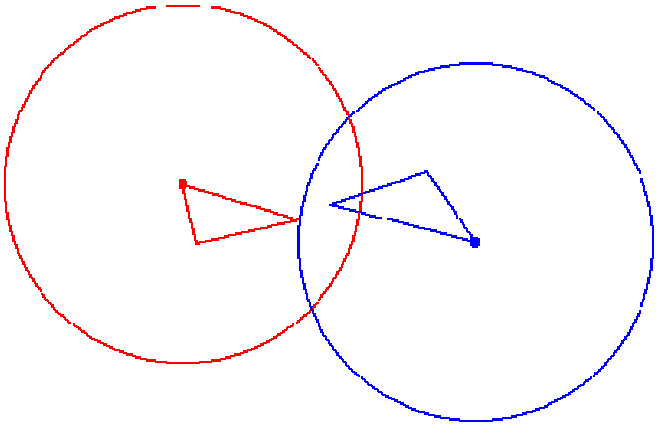

Figure 4 : Hausdorff distance shown around extremum of each triangles of fig. 1. Each circle has a radius of $H(P_1, P_2)$.

The other concern was the insensitivity of the shortest distance to the position of the polygons. We saw that this distance doesn't consider at all the disposition of the polygons. Here again, Hausdorff distance has the advantage of being sensitive to position, as shown in fig.5.

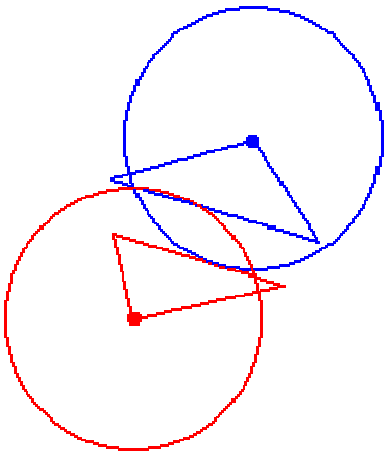

Figure 5 : Hausdorff distance for the triangles of fig. 4 at the same shortest distance, but in different position.

aus: “Hausdorff Distance between convex polygons”, 2005,
http://www-cgrl.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html