Perverse and Non-Perverse Geometry: from Hausdorff Distance to GPU

Klara Kedem

Computer Science Department
Ben-Gurion University of the Negev

SharirFest May 23, 2010
The title was inspired by an early 1990's quote distinguishing between non-perverse (pure) and perverse (applied) computational geometry.

Applications can dominate a field, especially computational geometry which has so many of them, such as in computer games, bioinformatics, computer graphics, image processing and on and on.

But the pure intellectual contributions, for which Micha is known, are immeasurably more influential and lasting
Still... motivated by problems from real life

We develop new tools that add to the theoretical knowledge base of computational geometry

We apply this knowledge in the application area (both engineering and scientific) to get new insights there as well
Perverse current research

Protein structure analysis (not in this talk)

RNA structural search (not in this talk)

More Bioinformatics (not in this talk)

Computer vision applications and GPU

Historical document analysis
It didn’t start like this: Micha tried to keep me on a decent track

Listen how he lured me to theoretical computational geometry
The beginning of my PhD with Micha

I came to Micha at an advanced age and said “I want a PhD in something practical”

He agreed, “do robotics”

The outcome was highly theoretical work with the opportunity to program at NYU robotics lab for about a year

Robotics not in this talk
Shape resemblance

The minimum Hausdorff distance started from a good smelling coffee emanating from Huttenlocher’s office at Cornell University....

While I was sipping the coffee he asked me an image comparison question: When can we say that two images are similar?

How do we measure this similarity?
What do we mean when we say that two shapes are similar?
Can we automatically track the moving car in these video pictures?
In each picture we have two 3D dendrites. One is **brown** and one is **blue**.

In what picture the brown and blue dendrites seem more similar?
Is there a similar binding site in the two proteins?
Intuition why the definition of resemblance is different

For the dendrites we squint

For the cars example we want to move one car onto the other and squint

For the proteins we want a 1-to-1 backbone match
The minimum Hausdorff distance for image comparison

The Hausdorff distance between pointsets A and B
The distance between point \( a \) in \( A \) and set \( B \)

\[
d(a,B) = \min_{b \in B} \text{dist}(a,b)
\]
The distance between sets $A$ and $B$

$$h(A,B) = \max_{a \in A} d(a,B)$$

(where: $d(a,B) = \min_{b \in B} \text{dist}(a,b)$)

This distance is not symmetric $h(A,B) \neq h(B,A)$
The symmetric Hausdorff distance

\[ H(A, B) = \max (h(A, B), h(B, A)) \]

The minimum Hausdorff distance under translation \( t = (t_x, t_y) \)

\[ H(A, B+t) = \max (h(A, B+t), h(B+t, A)) \]

\[ D(A, B) = \min_t (H(A, B+t)) \]

Putting \( t \) into the equations above... concentrating on \( h(A, B+t) \)
The Minimum Hausdorff distance as a function of $t$

$$h(A,B+t) = \max_{a \in A} d(a,B+t)$$
$$d(a,B+t) = \min_{b \in B} \text{dist}(a,b+t)$$

$$\text{dist}(a,b+t) = \text{dist}(a-b,t) \quad (\text{subtracting } b)$$

$$f_{a,b}(t) = \text{dist}(a-b,t)$$

Illustrated for points on a line
The Minimum Hausdorff distance as a function of $t$

$$h(A,B+t) = \max_{a \in A} d(a,B+t)$$

$$g_a(t) = d(a,B+t) = \min_{b \in B} \text{dist}(a,b+t)$$

$$f_{a,b}(t) = \text{dist}(a-b,t)$$
The lower envelope of cones in the plane is the Voronoi surface of the set $a-B$

$$h(A, B+t) = \max_{a \in A} d(a, B+t)$$
$$g_a(t) = d(a, B+t) = \min_{b \in B} \text{dist}(a, b+t)$$
$$= \min_{b \in B} f_{a,b}(t)$$

The lower envelope graph $g_a(t)$, for each $t$, is its distance to the closest point in $a-B$
The upper envelope of Voronoi surfaces

\[ h(t) = h(A, B+t) = \max_{a \in A} d(a, B+t) \]

The upper envelope in brown

The \textbf{min} Hausdorff distance under translation: Find \( t^* \) that minimizes \( h(t) \)
The upper envelope of Voronoi surfaces (cont.)

Notice that each local minimum on the upper envelope is an intersection point of two Voronoi surfaces.

This observation led to the cubic combinatorial bound on the number of local minima, or $O(mn(m+n))$ if $|A| = m$ and $|B| = n$. 
I have been talking on point sets in the plane.

Here is how one lifted Voronoi diagram looks (egg carton).
In 3D – the Voronoi regions are convex Polyhedra WAY more complicated

To compute the equivalent of pairwise intersections of Voronoi cells we have to find the boundary of the union of the convex polyhedra that are Voronoi cells of a point $q$ in pairwise Voronoi diagrams
A new bound was found (thank you Micha!) for the complexity of the boundary of the union of $s$ convex polyhedra with a total of $r$ faces, all polyhedra containing a common point $q$ --- $O(sr \alpha(sr))$.

And the algorithms followed in a straightforward way
Thus the initial practical question of image comparison turned into “application” when Micha joined us in “The upper envelope of Voronoi surfaces and its applications” DCG’93
The first almost-parallel computation of the minimum Hausdorff distance (rasterized)

In 1993 I met Danny Cohen-Or at BGU and together we designed a transformation of some of the theoretical algorithms onto a simple graphics processor that was available at that time.
The graphics operations “shift”, “and”, “or” and “max (z-buffer)” were the only available operations. The cone of $f(t)$ was “shifted” to points $a - b_i$, for all $i$ in $B$, an “or” operation yielded the Voronoi surface of one $a - B$, and the “and” operation yielded the maximum – the upper envelope of all the Voronoi surfaces.

We realized that for computer vision problems one needs a partial match – we did not have this capability yet.
The **partial** minimum Hausdorff distance (rasterized)
100% of the blue points are “close” to red points
80% of red points are close to green points

The definition of “closeness” depends on the user
The “closeness” and the “percentile” solved the “squinting” issue - when we squint we don’t care if some points in one image are really far from the other image (outliers) or if matching points are exactly on top of each other.

The dendrites below are from my work with Matya Katz and our student at BGU.
13 years later with Dror Aiger, we consider

The **Point matching (PM)** problem as the minimum Hausdorff distance decision problem

The **Largest Common Pointset (LCP)** problem as finding the largest common pointset among two pointsets under the Hausdorff distance (**allowing outliers**)
Our contribution

Exact and approximate \textbf{PM} under similarity in $O(mn^2 \log n)$ and $O(n^2 \text{polylog}(mn))$ respectively ($m<n, L_\infty$, separability restrictions)

Approximate \textbf{LCP} using the GPU (graphics processing unit) for rigid, homothetic and similarity transformations

Input Sensitive approaches:

\textbf{PM} in the plane under rigid, homothetic and similarity in roughly $O((n+km) \log n)$

Fast in practice \textbf{PM} under any transformation for points in any dimension by a Branch and Bound input sensitive algorithm
From matching to depth in an arrangement

The transformations that map a point \( p \) in \( P \) close to a point \( q \) in \( Q \) form an object in transformation space (for rigid transformation, we approximate arcs by segments to the desired accuracy)

\[ a, b, c, \] are the parameters of the transformation

Points \( p \in P \) and \( q \in Q \) in the plane
From matching to depth in an arrangement (cont.)

We can assume that the $n \delta$-neighborhoods of points in $Q$ are disjoint in their interiors otherwise we can decompose them to $O(n)$ disjoint rectangles.

The point in transformation space covered by the maximum number of objects transforms the maximum number of points from $P$ close to $Q$. 

An object in transformation space

$\delta$-neighborhood of $q \in Q$

Points $p \in P$ and $q \in Q$ in the plane
Basic GPU operations

A reduction to depth in arrangement of objects in low dimensional transformation spaces

Approximating the maximum depth:
- 2D trans. space: incremental polygon fill
- 3D trans. space: “peeling” levels one by one using the GPU and counting objects
- 4D lowering the dimension of trans. space to 3

Reducing the number of objects by randomization and GPU based intersections
Peeling levels by GPU

Given a set $C$ of hyperplanes, a point $p$ on hyperplane in $C$ is said to be at $k$-level, if there are exactly $k$ hyperplanes in $C$ lying strictly below $p$. 

Arrangement of geometric objects

0-level

Maximum depth
From k-levels to depth by GPU

In an arrangement of convex objects, the depth is the number of objects that a ray down to infinity crosses only once.

The depth can be computed by going through the k-levels, one by one and counting the number of “going in” and the number of “going out” events, for all pixels in parallel.
Historical document analysis
With Itay Bar Yosef and Tsiki Dinstein

- Historical documents degrade over time:
  - Smudges
  - Foreground and background are hard to separate (bleed-through effect)
  - Broken and deformed lines and characters
Our goals are

Dealing with very deteriorated documents

Binarization

Page and line segmentation

Character and word spotting
Animation of binarization

Original → global thresholding → seed → growing regions
Text line segmentation and dewarping will aid in word (character) spotting/recognition and more.
Line segmentation
Dewarping
Segmentation of highly degraded characters with adaptive shape priors
Many of the algorithms for historical documents have also been implemented in GPU by switching from control of iterations in CPU to matrix computations in GPU.

TESLA is the new NVIDIA General Purpose GPU (GPGPU), having up to ~2000 processors.