Voronoi diagrams and applications

Prof. Ramin Zabih

http://cs100r.cs.cornell.edu

Cornell University
Computer Science

Administrivia

- Last quiz: Thursday 11/15
- Prelim 3: Thursday 11/29 (last lecture)
- A6 is due Friday 11/30 (LDOC)
- Final projects due Friday 12/7
- Course evals!

  http://www.engineering.cornell.edu/courseeval/

  - This will count towards your grade!
    - We get a list of students who didn't fill it out, and will contact you

Final projects

- All the proposals were fine
- Due Friday, Dec 7, 4-5PM
  - Pizza will be provided!
  - A few “I survived CS100R” T-shirts
- Other CS faculty will likely come by
  - There may also be photographers, etc.

Image differencing

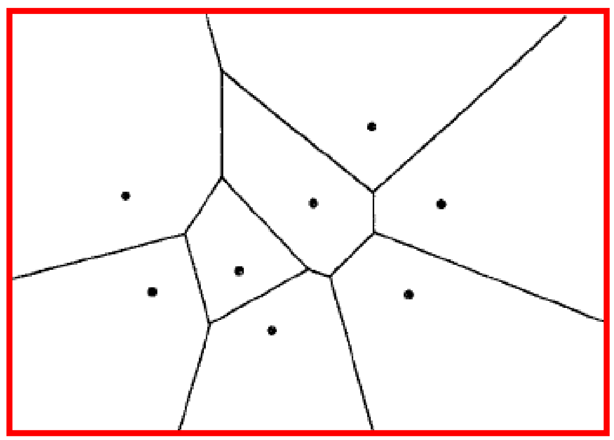
Figure from Tim Morris

Applying image differencing

- By thresholding the difference image, we can figure out which pixels belong to the new object
  - I.e., the Coke or Pepsi can
- These pixels will have some colors
  - They are points in color space
- More generally, Coke pixels will tend to form a group ("cluster") that is largely red
  - Pepsi will be largely blue
- We can then compute a few model pixels

Voronoi diagrams

- Divide our space into regions, where each region reg(P) consists of the points closest to a labeled point P
  - This is a Voronoi diagram
  - Long history (Descartes, 1644)



Impossible algorithms, redux

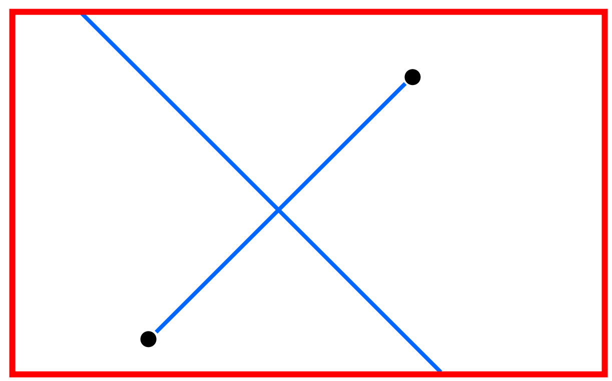
- There are no O(n) sorting algorithms
  - More precisely, none based on comparisons
- You can use convex hull to sort
  - By placing the points on a parabola
  - So, is there an O(n) convex hull algorithm?
- You can use Voronoi diagrams to compute a convex hull
  - So, is there an O(n) Voronoi diagram algorithm?
    - You can also use 3D convex hull to compute a 2D Voronoi diagram

Using Voronoi diagrams

- Two obvious questions:
  - How can we efficiently create it?
  - How can we use it, once we've got it?
- A Voronoi diagram divides the space into Voronoi cells, reg(P) for some P
- If reg(P) is a strange shape, hard to figure out if the query is inside reg(P)
  - Fortunately, as the picture suggests, Voronoi cells have simple shapes

Computing reg(P)

- Consider some other labeled point Q
- Points might be in reg(P) if they are closer to P than to Q



  - I.e., they are in a polygon (half-plane)
- reg(P) is the intersection of N-1 polygons
  - There are faster ways to compute it

Voronoi cell properties

- The polygons whose intersection is reg(P) have another important property
  - They are convex!
- The intersection of two convex shapes is also a convex shape

Voronoi query lookup

- Given a Voronoi diagram and a query point, how do we tell which cell a query falls into? (I.e., solve the 1-NN problem)
- We can project down to the x-axis every point in the Voronoi diagram
  - This gives us a bunch of “slabs”
  - We can find which slab our query is in by using binary search
  - Within a slab, we can find the Voronoi cell using binary search
  - Unfortunately, this is pretty messy

Example of slabs

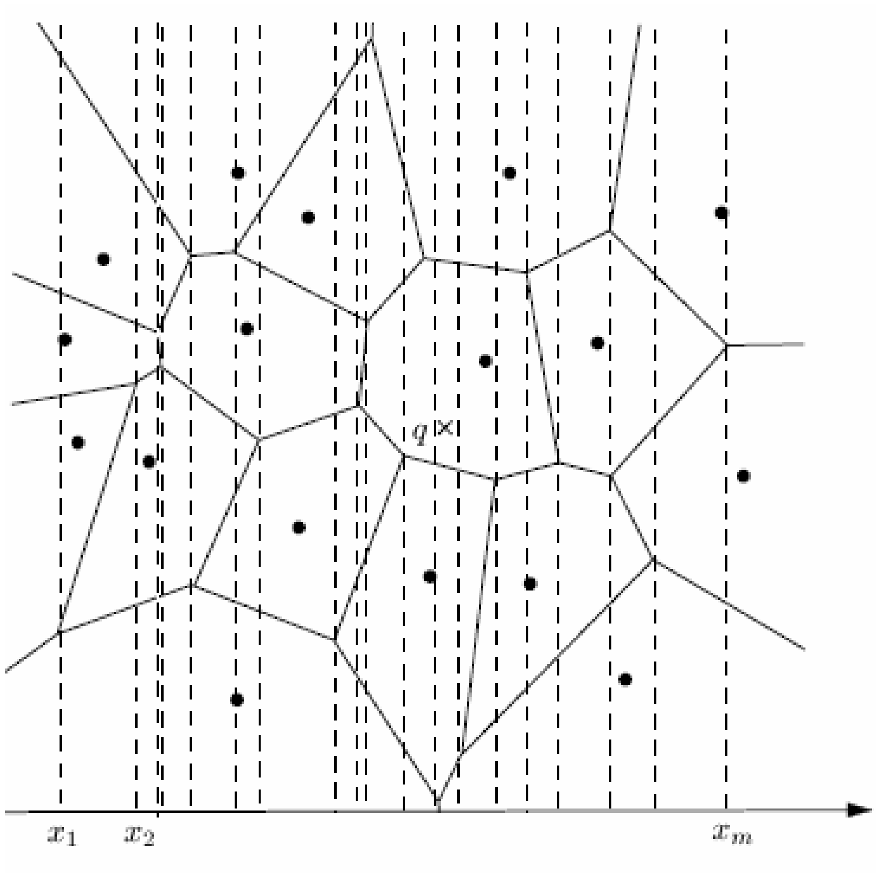
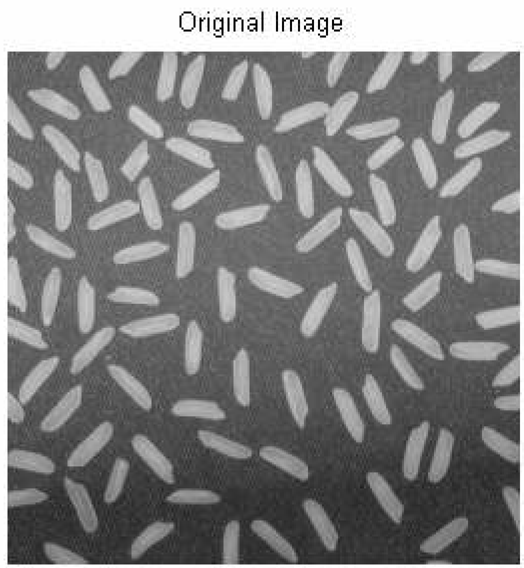


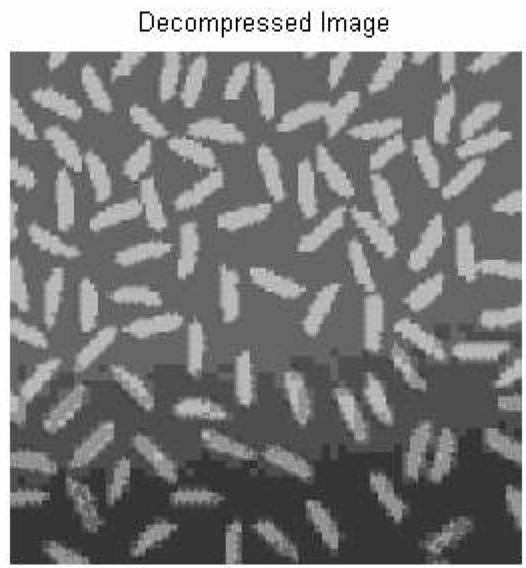
Figure from H. Alt “The Nearest Neighbor”

A Voronoi application

- Image compression: generate a terse representation of the image by allowing small errors to be introduced
- Simple method: vector quantization (VQ)
  - Video applications: Cinepak, Indeo, etc.
    - Also used for audio (Ogg Vorbis)
  - In a color image, each pixel has a 24-bit number associated with it (the colors)
  - We can generate a “code book” with, say, 2^8 entries, and use this instead of the colors

Example output (Matlab)





Triangulations

- Suppose that we want to analyze a curved surface, such as a wing or a vase
  - We can approximate it by a lot of small low-order polygons, especially triangle
  - This is tremendously important for, e.g., building planes, bridges, or computer graphics
    - Finite Element Method!
- Want to build a triangulation out of fat triangles, not skinny ones
  - Using the Voronoi diagram, we can generate a high-quality triangulation!

Delaunay triangulation

- Connect two input points if they share an edge in the Voronoi diagram
  http://www.cs.cornell.edu/Info/People/chew/Delaunay.html
- This maximizes the smallest angle
- The circumcircle of a triangle passes through all 3 of its vertices
  - Not the same as the bounding circle
  - In the Delaunay triangulation, the circumcircles contain no other points

Applications of triangulation