Möbius-Invariant Natural Neighbor Interpolation

Marshall Bern

Palo Alto Research Center

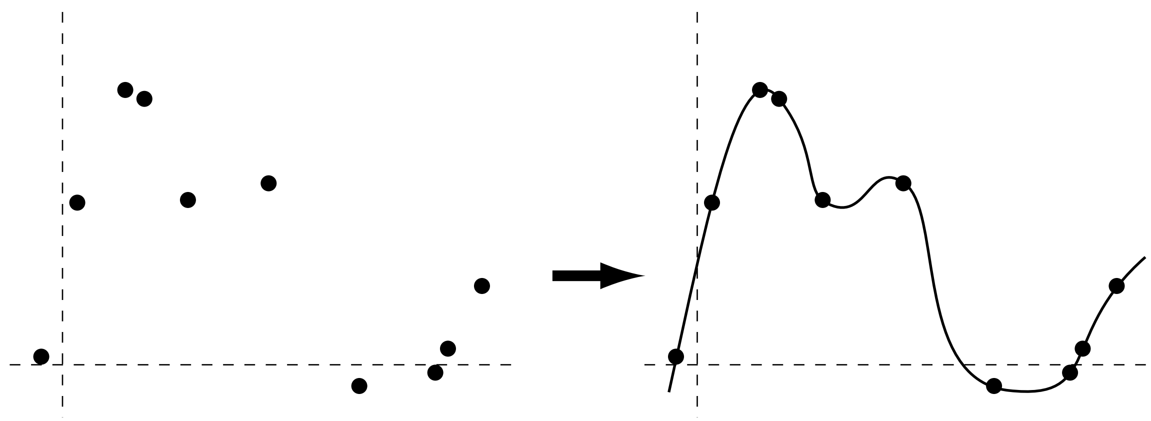
David Eppstein

Univ. of California, Irvine
School of Information and Computer Science

What is interpolation?

Reconstruct a function (approximately)
given a discrete set of samples
(function values at finitely many data points)

Should exactly fit samples, be well behaved elsewhere



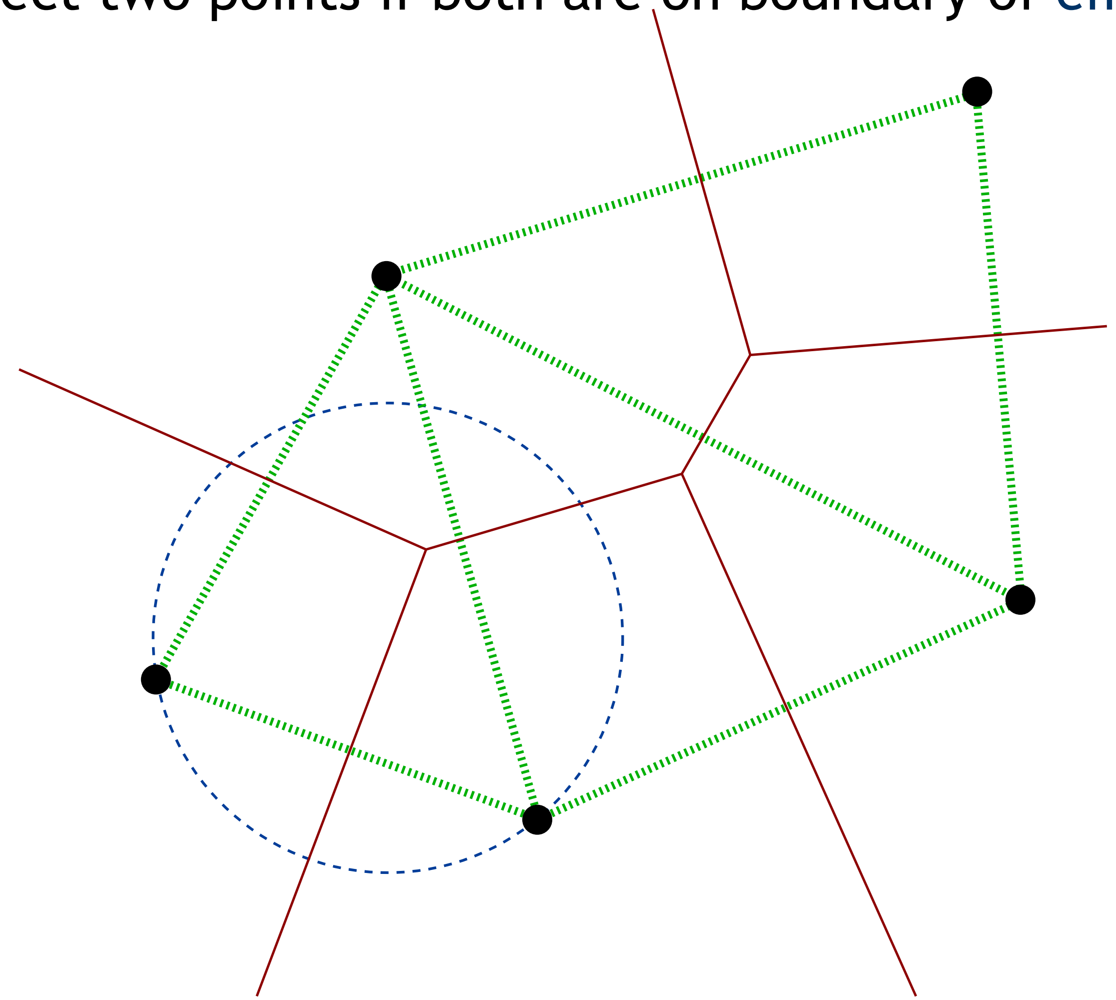
Data may form regular grid or irregular scattered data
here we consider irregular data in two dimensions

Many interpolation algorithms known...

What are natural neighbors?

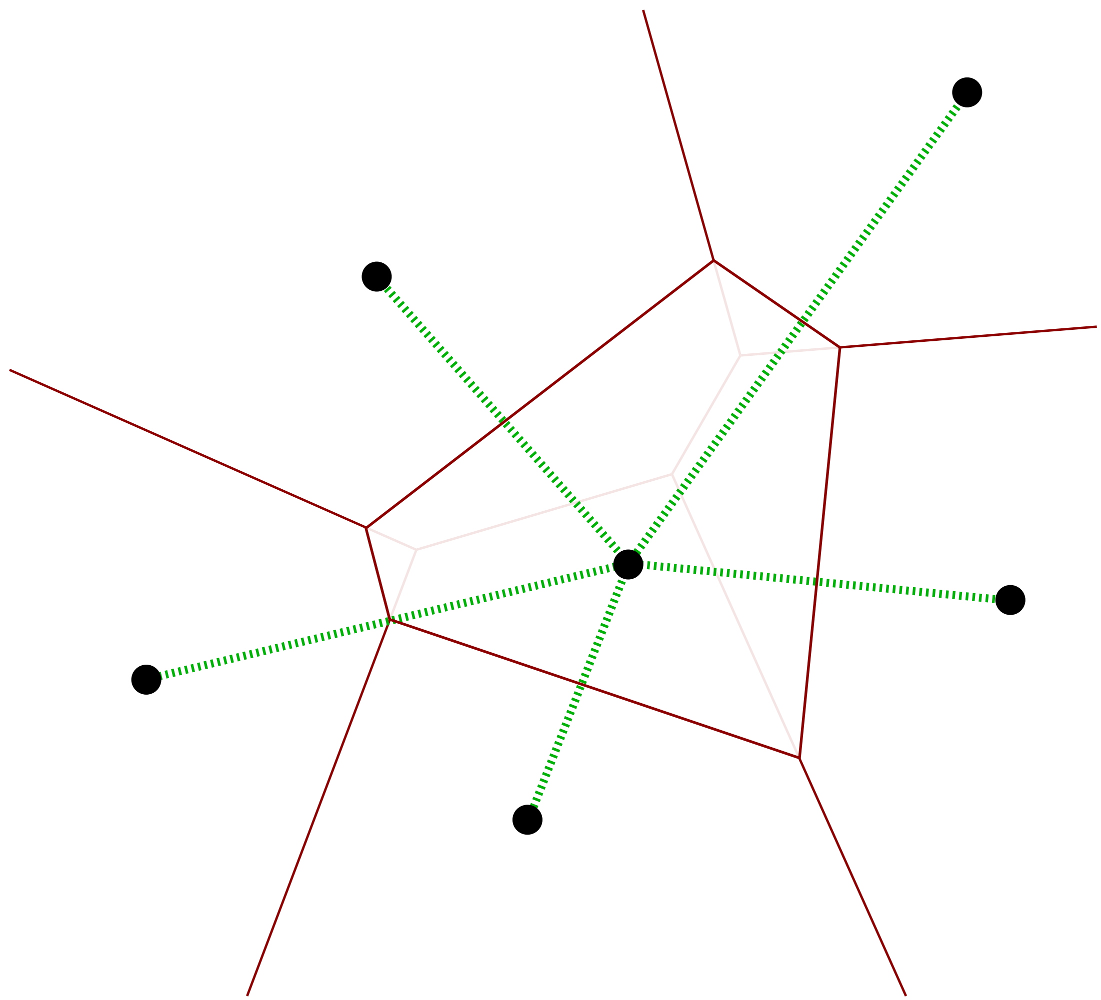
Voronoi diagram:
partition plane into cells nearest each data point

Delaunay triangulation:
connect two points if both are on boundary of empty circle



What are natural neighbors?

Natural neighbors of point *x*:
insert *x* into Delaunay triangulation of sample points
find neighbors of *x* in augmented triangulation



Neighbor-based interpolation

To compute interpolated function at point *x*
find set of neighbors of *x*, weights for each neighbor

Interpolated value = weighted average of neighbor values
Invariant under affine changes of function value

One nearest neighbor, weight = 1:

Interpolated function is constant in each Voronoi cell
But discontinuous on Voronoi boundaries

Used for rainfall estimation

Neighbors = corners of Delaunay triangle containing *x*
weights = barycentric coordinates in triangle

Interpolated function is linear in each triangle
But non-smooth on Delaunay edges

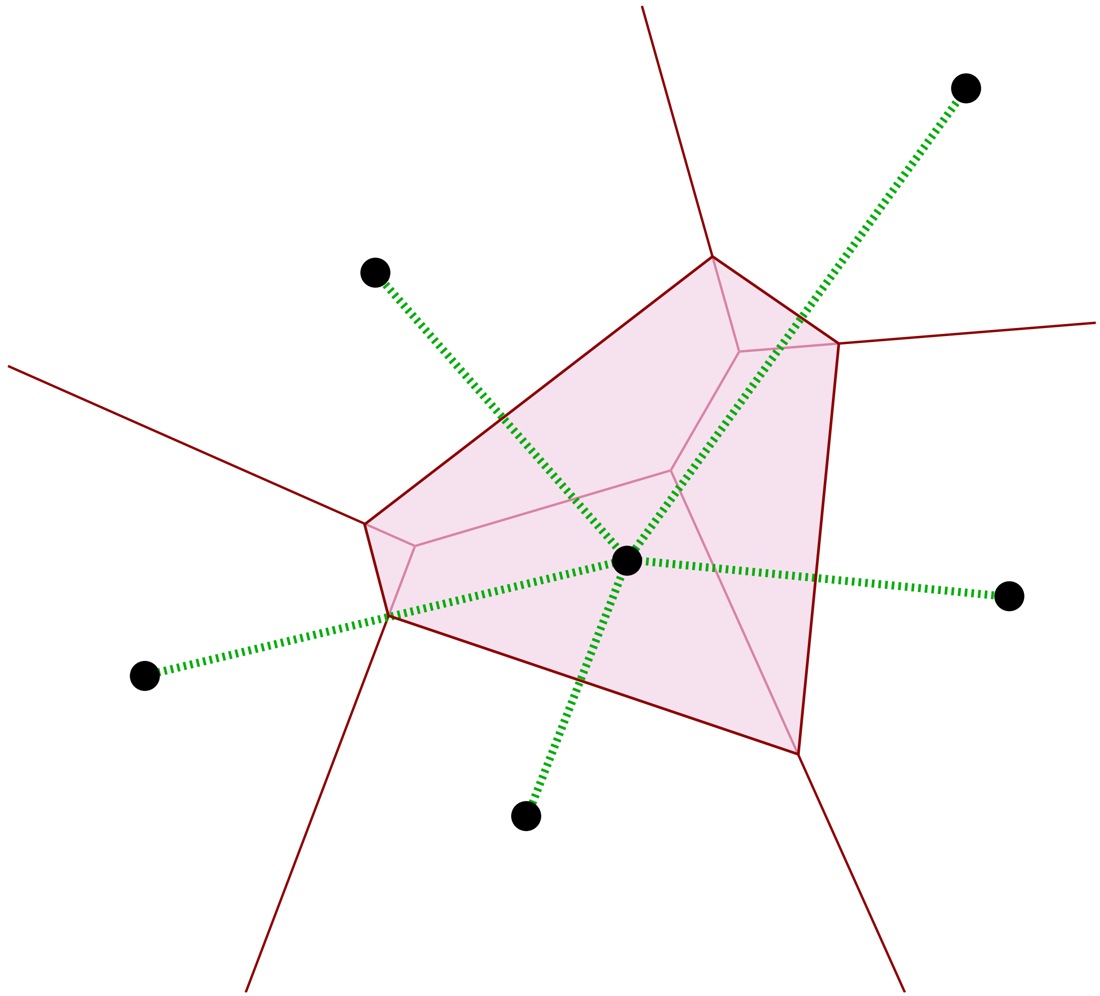
Used for earth surface reconstruction

Natural neighbor interpolation

[Sibson, 1981]

Neighbors = natural neighbors

Weight(*y*) = area of *y*'s Voronoi cell covered by new cell for *x*

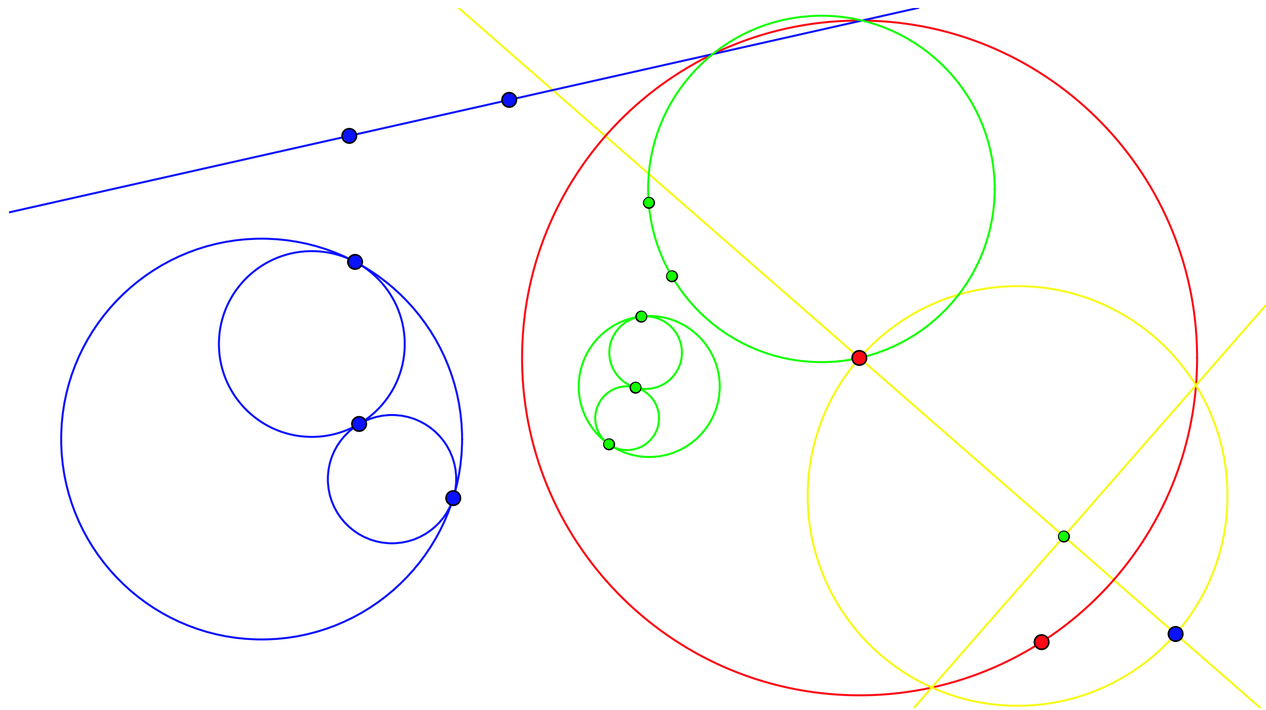


Continuous, smooth except at sample points

Correctly reconstructs linear functions

Inversion

Given any circle (red below)
map any point to another point on same ray from center
product of two distances from center = radius^2



Circles ⇔ circles
(lines = circles through point at infinity)

Conformal (preserves angles between curves)

Möbius transformations = products of inversions

Forms group of geometric transformations
Contains all circle-preserving transformations

In higher dimensions (but not 2d) contains all conformal transformations

Previous work [Bern and Eppstein, WADS 2001]
on finding Möbius transform optimizing transformed shape

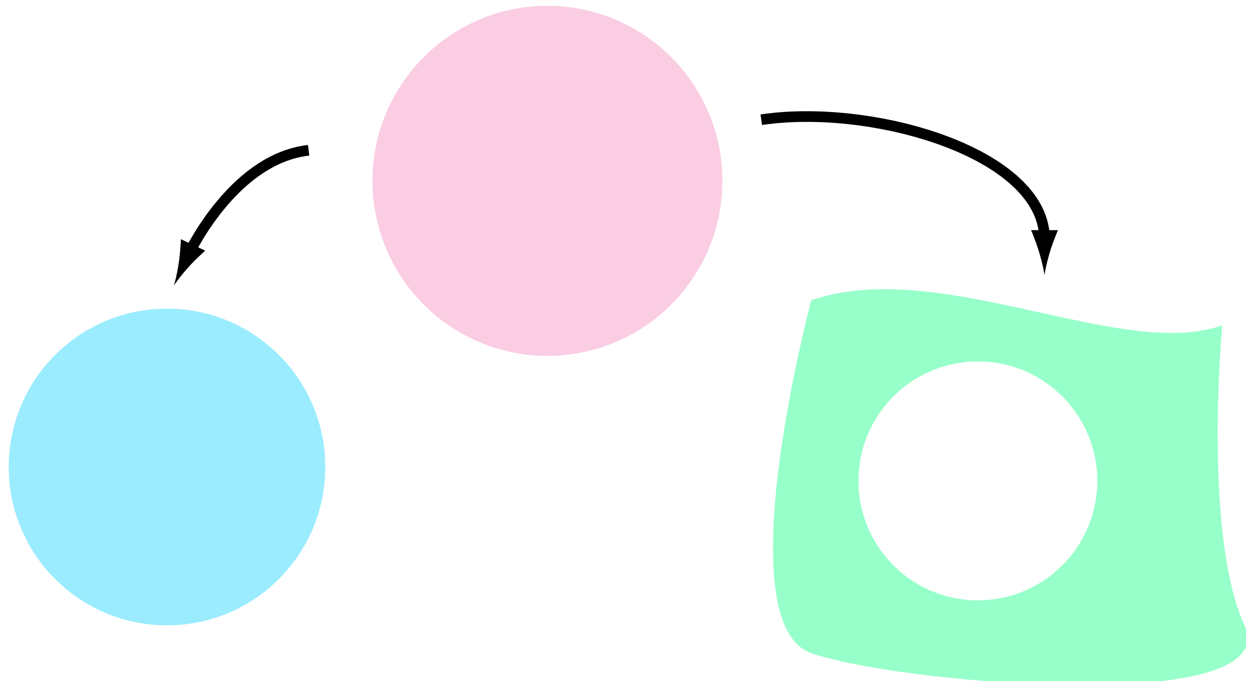
Can we find a Möbius-invariant interpolation algorithm?

interpolate(transform(data)) = transform(interpolate(data))

must be continuous at infinity, so can't reconstruct linear functions
Harmonic functions invariant under Möbius transformation, reconstructable?

Möbius transformation of natural neighbors

Empty circle is transformed to another empty circle
or to empty complement of circle



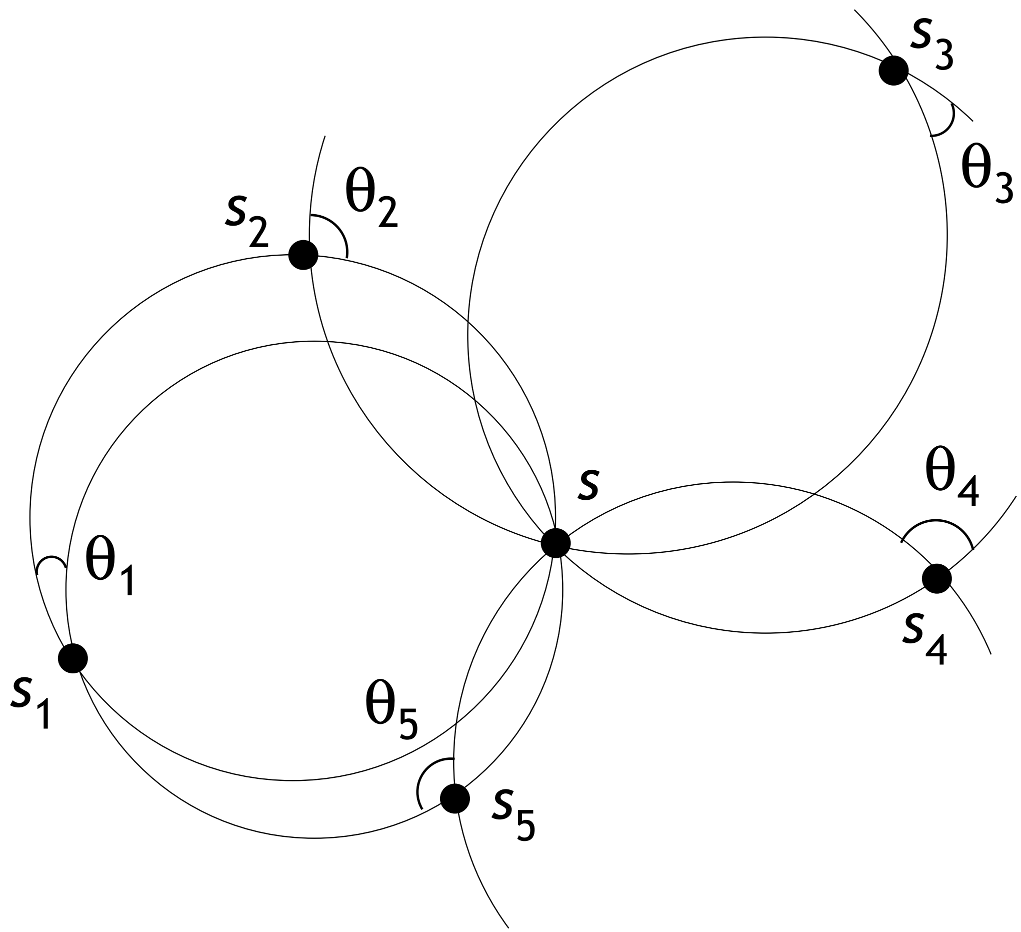
Extended natural neighbor = point on boundary of empty circle or complement
= neighbor in augmented DT or augmented farthest-point DT

Set of neighbors is invariant under Möbius transformation

so, natural to seek Möbius-invariant natural neighbor interpolation...

What to use for weights?

Voronoi areas not invariant under Möbius transformation
Instead, use functions of angles between Delaunay circles



Alternative interpretation of angles:
Transform plane so interpolated point goes to infinity
Use angles of convex hull of transformed samples

What function of Delaunay angles to use for weights?

As interpolated point approaches data sample, sample's angle $\to \pi$

So in order to continuously interpolate the sample,
need $w(\theta) \to \infty$ as $\theta \to \pi$

Unable to exactly reconstruct harmonic functions from finite data
(function space too high dimensional)
Instead, reconstruct in limit of dense samples on circle

Harmonic measure on circle (as viewed from center) = arc length

So, in order to approximately reconstruct harmonic functions,
need $w(\theta)/\theta \to$ constant as $\theta \to 0$

Natural choice satisfying both constraints: $w(\theta) = \tan(\theta/2)$

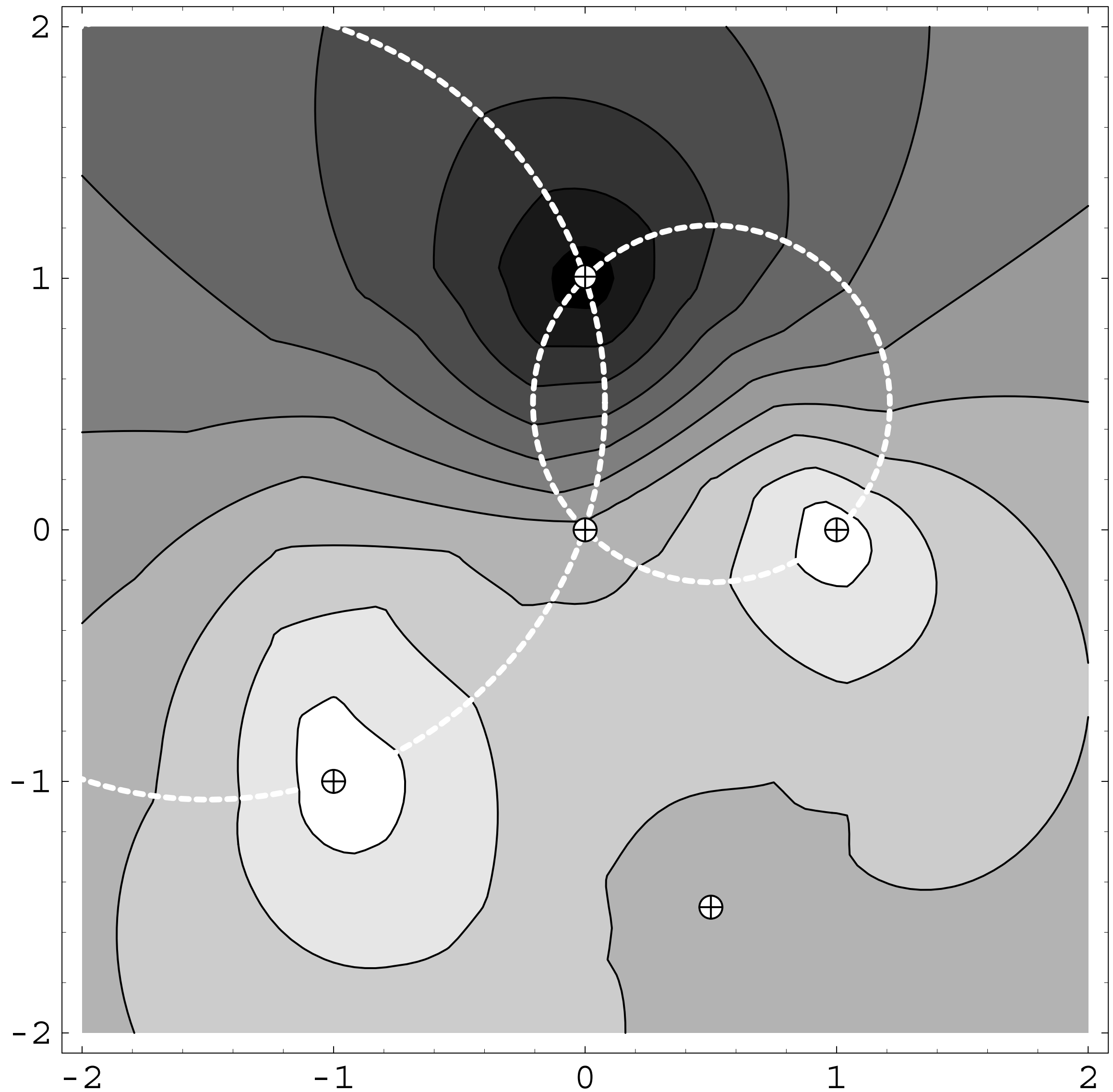
The Main Results

Use neighbor-based interpolation
With neighbors = extended natural neighbors
Weights = tan(Delaunay circle angle / 2)

(1) Result is a continuous function
interpolating the sample data

(2) Let f be any harmonic function on a closed disk
and let ε = maximum distance between samples on disk boundary.
Then as $\varepsilon \to 0$, the interpolation converges to f.

Contour plot of example interpolation



Note lack of smoothness along Delaunay circles...

Time Bounds

Time to interpolate a single point: O(n log n)
(transform, take convex hull)

Time to compute whole diagram: O(n^2)
(form arrangement of O(n) Delaunay circles)

Conclusions

Showed how to define natural neighbors in a Möbius-invariant way

Found angle-based weights for neighbors
such that neighbor-based interpolation is:

continuous, correctly interpolates sample points

approximate reconstruction of Harmonic functions

Open Questions

Our interpolation is not smooth
Is there a natural choice of smooth Möbius-invariant interpolation?

What about higher dimensions?