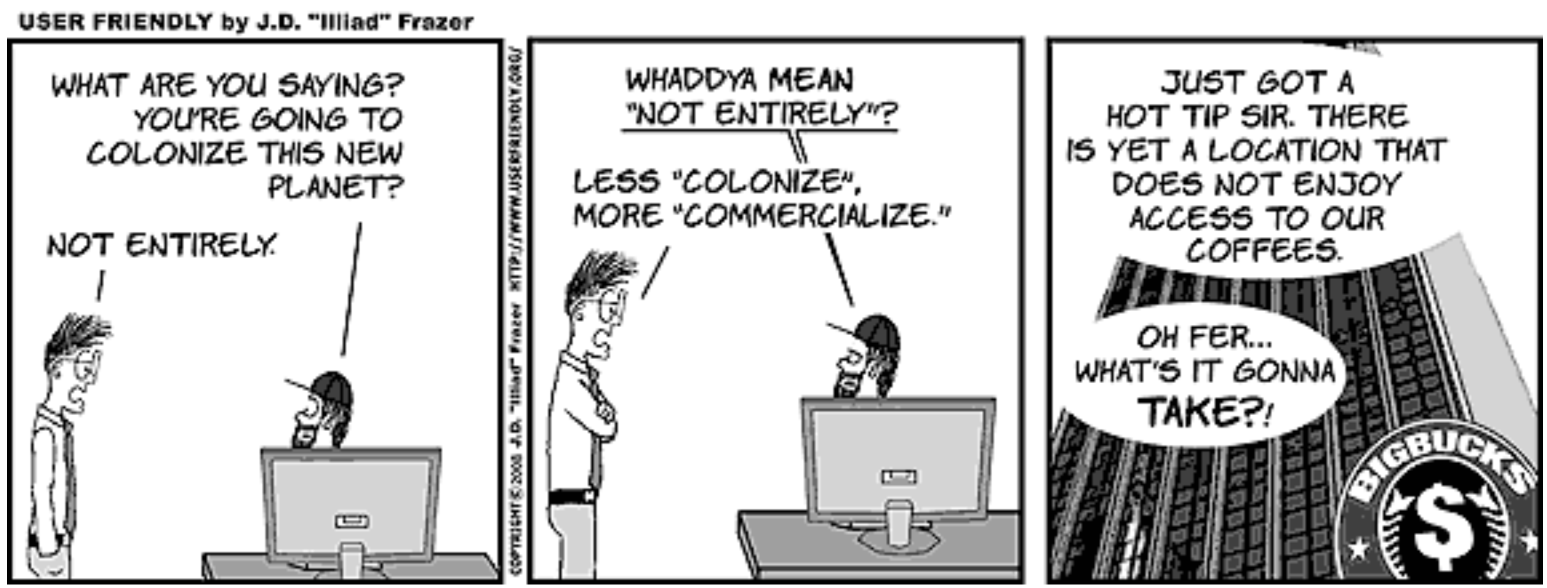
Scattered Data Interpolation

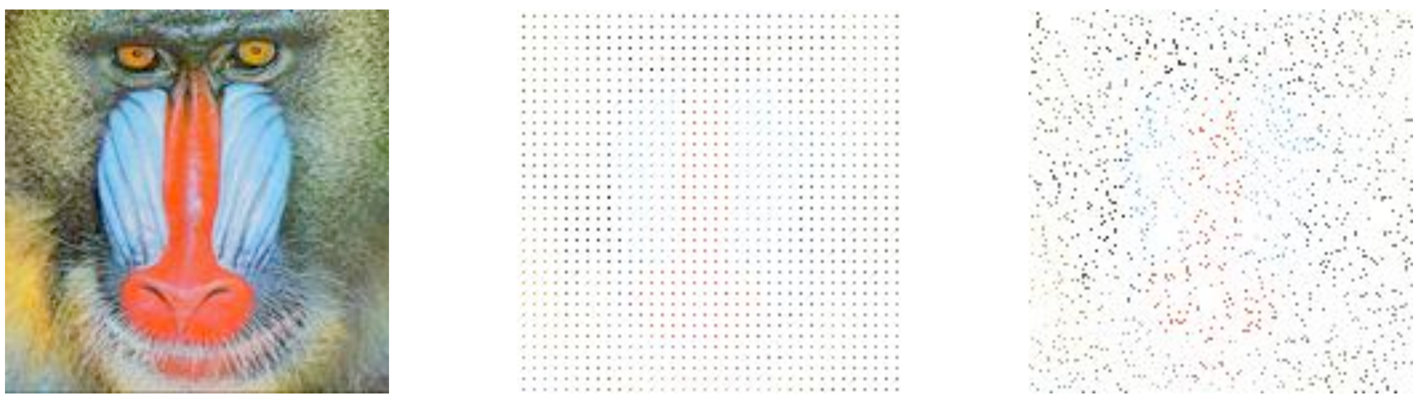


http://www.userfriendly.org/

Problem set #4 will be posted tonight

Problem Statement

- When we previously explored reconstructing a signal from a sparse set of samples, we assumed that those samples were uniform (a.k.a. periodic).
- Today we consider methods for constructing images given nonuniform or scattered samples



Two sampled versions of the original Mandrill image. Both retain 1/16th of the original's pixels.

Revisiting Reconstruction

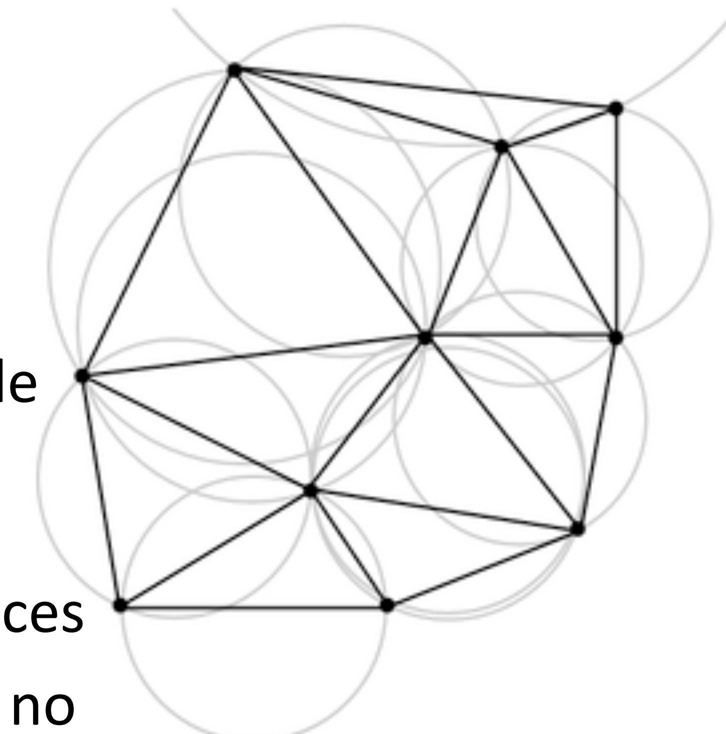
- Central Idea – to find the best continuous signal approximation that satisfies the constraints of the given samples
- Constraints can come in different flavors
  – Interpolating: the signal must pass through the samples
  – Approximating: the signal should minimize its difference from the given samples, while satisfying other constraints such as smoothness, polynomial degree, degree of continuity
- Samples can be and measure; the signal's value, derivative, local tangent space, etc.

A C_0 Continuous Interplolation

- Triangulations are a natural choice for reconstructions
- Signal values are treated as vertex features that are interpolated within the triangle's interior
- Piecewise planar reconstruction
- Intersection of signal and geometry processing
- However, not all triangulations are reasonable
  - No overlap: Each domain point falls within a single triangle
  - Locality: triangles should somehow be limited to nearby points

Candidate Triangulations

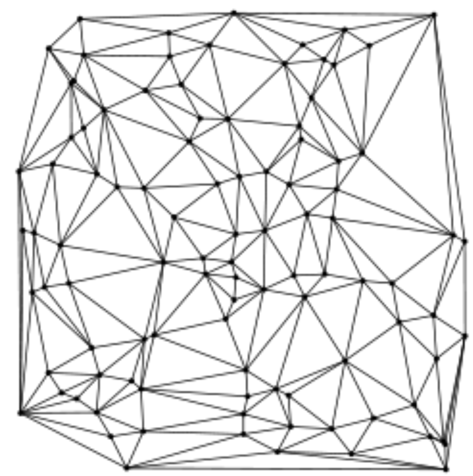
- Delaunay Triangulations (DT)
  - A triangulation is a list of (D+1)-tuples of vertices, where D is the dimension of the signal
  - For each non-degenerate triangle there exists a unique D-dimensional circumsphere passing through each of its vertices
  - The triangulation is Delaunay iff no triangle in the list has any other vertex with its circumsphere



A 2-dimensional Delaunay triangulation showing each triangle's circumsphere

Delaunay Properties

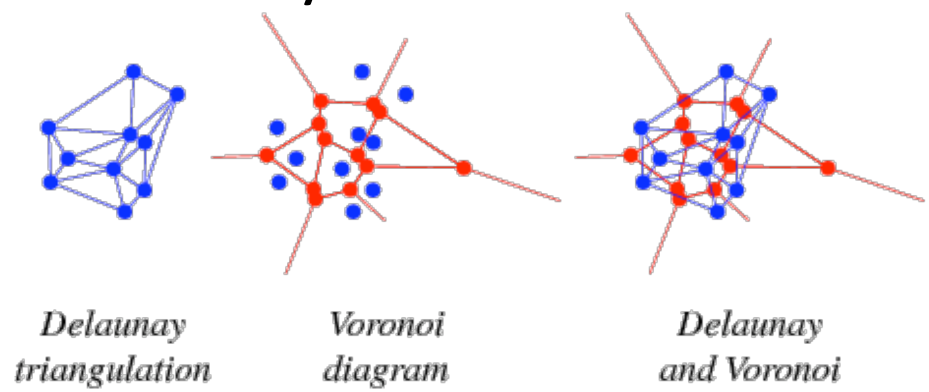
- No overlaps
- Convexity: The union of all Delaunay triangles gives the convex hull of the vertex set.
  This is a nice property to have for interpolation since it (combined with no overlaps) insures that all points interior to the hull will have a unique value



- In the plane, a DT maximizes the minimum angle. Compared to other triangulations, the smallest angle in a DT is at least as large as the smallest angle in any other. (i.e. Triangles are closest to equilateral vs any other triangulation choice)

Delaunay-Voroni Duals

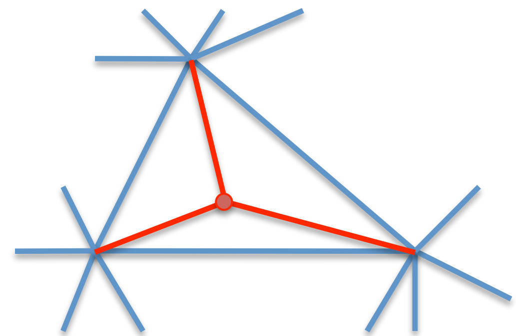
- Previously we discussed the notion of a Voroni cells for describing the nearest representative point over an entire domain (vector quantization)
- DT and Voroni Cells are intimately related
- Edges of a DT are bisected by VC boundaries



- They are "dual graphs"
  Ts→ Vs,
  adjacent T edges → Es

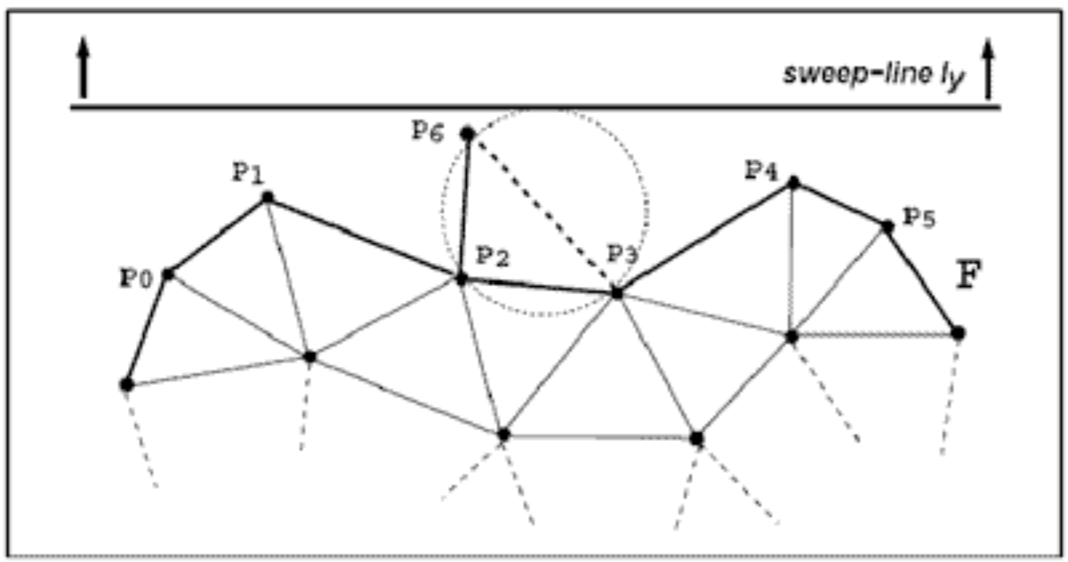
Delaunay Algorithms

- Incremental Flipping Approach
  - Starts with an initial triangulation the whole domain
  - Insert vertices, one at a time, and split the triangle that the vertex falls into
  - Then flip triangulations until all circumsphere properties are satisfied
  - "Usually" requires only local changes



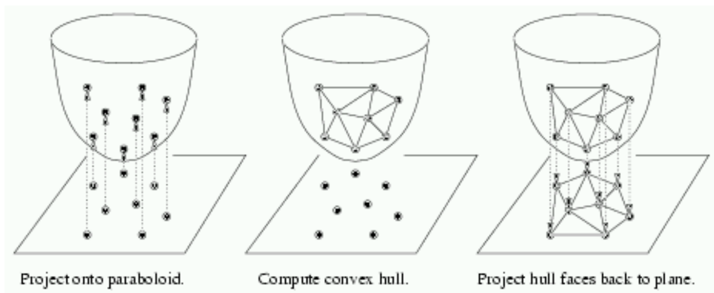
Another Algorithm

- Another algorithm that is particularly efficient, but limited to 2D, is the Fortune's "plane sweep" approach
- Sweep all points with a line
- Points behind the line are triangulated
- As the line sweeps other points new triangles are built.



One More DT Algorithm

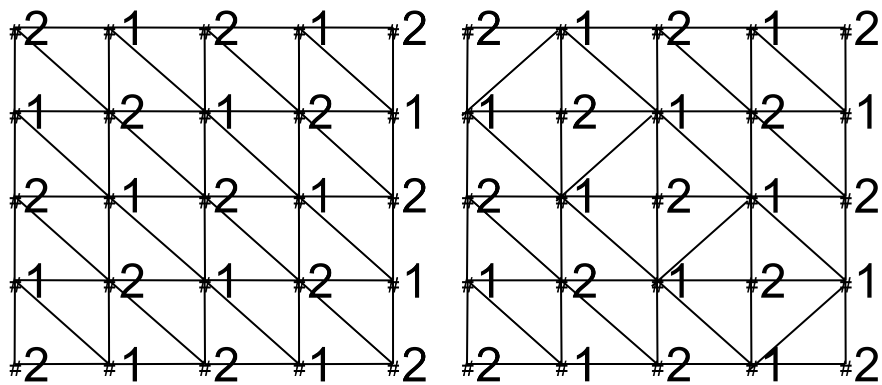
- One of the most robust methods for finding a DT is to find the convex hull of a D+1 paraboloid using a lifting approach, and then projecting its image back to D-dimensions.



- Carefully consider whether to implement your own DT algorithm. They are fraught with subtle numerical issues and special cases. First consider using a robust DT package (e.g. qhull)
- Still interested, take Comp 651 in the Spring

Other Triangulations

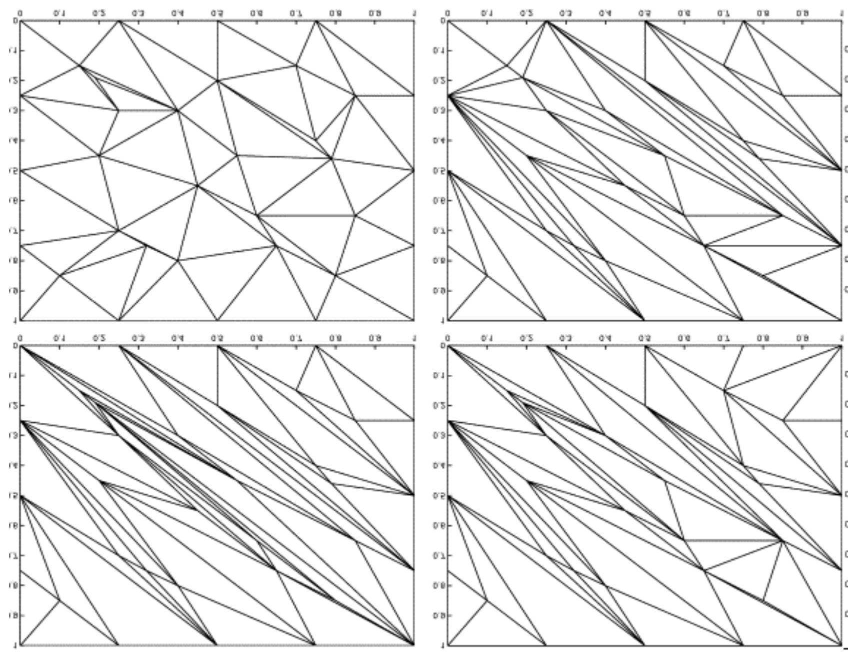
- DT is degenerate in the uniform case



- DT does not always give satisfying results
  - Triangulation is independent of signal being interpolated
  - In areas of rapid signal variation the choice of triangulation can have a large impact, particularly if the change has orientation

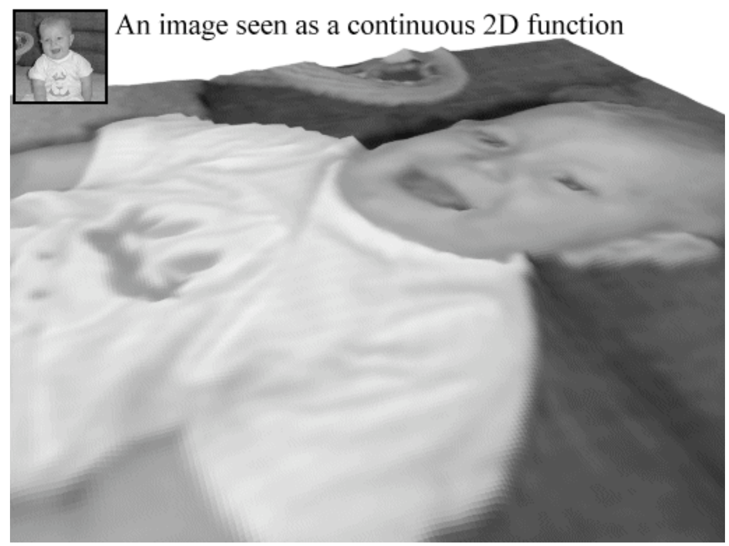
Data-Dependent Triangulations

- Introduced by Dyn et al (1990) and analyzed by Wang et al (2000) and Alboul *et al* (1999)
- Idea
  - Given a starting triangulation
  - Arrange Ts into a priority queue by some data-dependent metric
  - Greedily apply flips to improve local objective functions
    - » Minimize change in slope (angles between abutting normals)
    - » Minimize perimeters or curvature of isophotes



Basis Function Approaches

- For uniformly sampled data our approach to reconstruction was to place a continuous kernel (basis) function at each sample whose sum is the reconstruction
- Can we capture the essence of this idea and apply it to non-uniformly sampled data.
- What class of kernels should be used?
- Recall the property of a Kernel function that made it interpolate rather than approximate the signal?



Approximating Kernels

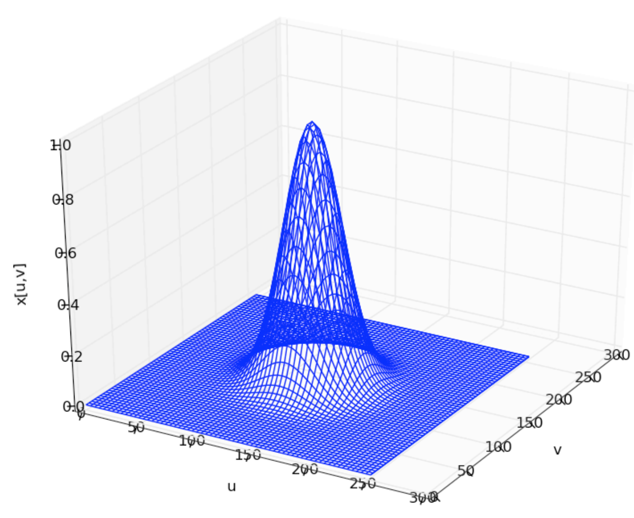
- Difficult to design well-behaved functions with unit responses at each sample, but roots at all others
- Prefer a regular kernel (same basic form at each sample), with relatively few parameters to optimize over
- Final functions inherit the continuity of their kernels

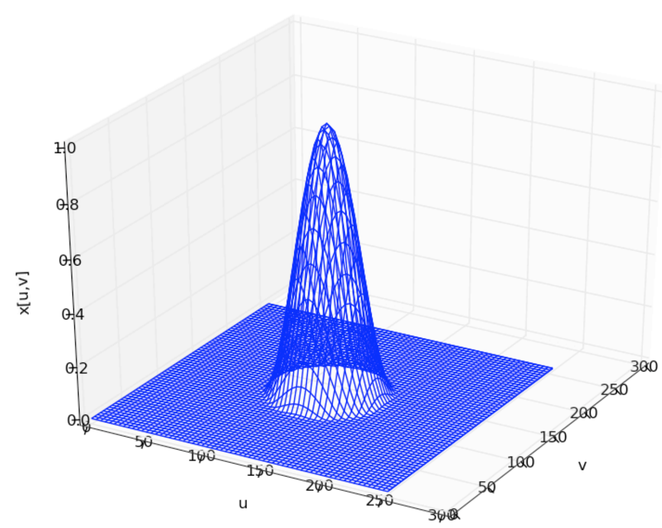
If $\Phi(x,y)$ is C_N, then, in general, $f(x,y) = w_1\Phi_1(x,y) + w_2\Phi_2(x,y) + \dots + w_n\Phi_n(x,y)$ will also be C_N (recall continuity is a function of the number of derivatives)

Where $\Phi_i(x,y)$, are all the same function translated to the sample position and scaled by the weighting factor w_i

Kernel Types

- Global – Kernel with infinite support, like Gaussians
- Local – Kernels with a finite no zero region, like a raised cosine, and uniform cubic b-spline





Approach

- Form of ϕ_i

$$r_{u,v} = \sqrt{(\mathrm{x} - \mathrm{u})^2 - (\mathrm{y} - \mathrm{v})^2}$$

$$\phi_{u,v,\sigma}(x,y) = e^{-\frac{r_{u,v}^2}{\sigma}} \quad or \quad \phi_{u,v,\sigma}(x,y) = 0.5(\cos(\frac{r_{xy}\pi}{\sigma}) + 1)$$

- Given $f(x,y,\sigma) = w_1\Phi_1(x,y) + w_2\Phi_2(x,y) + \ldots + w_n\Phi_n(x,y)$
- Choose a fixed σ and solve for w_i's
- Leads to a linear system

Linear System

- N equations, N unknowns
- Matrix of constant values
  (each kernel function evaluated at the position of each sample)
- Choose the global constant σ so kernels overlap at worst case sample spacing (particularly important for kernels with finite support)

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} \phi_{u_1,v_1,\sigma}(u_1,v_1) & \phi_{u_2,v_2,\sigma}(u_1,v_1) & \cdots & \phi_{u_{n-1},v_{n-1},\sigma}(u_1,v_1) & \phi_{u_n,v_n,\sigma}(u_1,v_1) \\ \phi_{u_1,v_1,\sigma}(u_2,v_2) & \phi_{u_2,v_2,\sigma}(u_2,v_2) & & \phi_{u_{n-1},v_{n-1},\sigma}(u_2,v_2) & \phi_{u_n,v_n,\sigma}(u_2,v_2) \\ \vdots & & \ddots & & \vdots \\ \phi_{u_1,v_1,\sigma}(u_{n-1},v_{n-1}) & \phi_{u_2,v_2,\sigma}(u_{n-1},v_{n-1}) & & \phi_{u_{n-1},v_{n-1},\sigma}(u_{n-1},v_{n-1}) & \phi_{u_n,v_n,\sigma}(u_{n-1},v_{n-1}) \\ \phi_{u_1,v_1,\sigma}(u_n,v_n) & \phi_{u_2,v_2,\sigma}(u_n,v_n) & \cdots & \phi_{u_{n-1},v_{n-1},\sigma}(u_n,v_n) & \phi_{u_n,v_n,\sigma}(u_n,v_n) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix}$$

Surprising Kernels

- The best scattered data kernels in practice can be very counter intuitive
- More next time