Surface Reconstruction from Point Set Using Projection Operator

Ly Phan, Lu Liu, Sasakthi Abeysinghe, Tao Ju, Cindy Grimm

Problem

- Given a point set, construct a surface interpolating or approximating the input points that is not only topologically correct but also geometrically smooth.

Previous work

- Delaunay-based
  - –Power crust [*Amenta, Choi and Kolluri, 2001*]
  - –Tight Cocone [*Dey and Goswami, 2003*]
- Implicit functions
  - –Radial basis functions
  - –Moving Least Square [*Shen, O'Brien and Shewchuk, 2004*]
- Projection-based [*Alexa et al. 2003*], [*Levin 2003*], [*Alexa and Adamson 2004*].
- Smooth surface fitting
  - –Subdivision surface
  - –Constructive manifolds [*Grimm, Laidlaw, and Crisco, 2002*]

Challenges

- Approaches such as Delaunay-based and Implicit functions can only work with closed surfaces
- Projection-based approach does not construct an explicit surface.

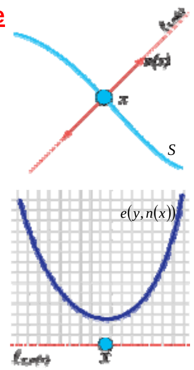
Our method

- Observation: Defined as an extremal surface by [*Amenta and Kil 05*], the point set surface can be considered as the singularity of an oriented vector field, which can be computed directly using a contour-like approach.
- We propose a grid-based algorithm that constructs an explicit surface given a smooth vector field and an energy function.
- We also propose a smooth vector field based on distance field (gradient of distance field), that allows our algorithm to generate surfaces with boundaries.
- Our method requires neither extra information of the data point set nor prior knowledge of the surface.

Defining Point Set Surface

- The definition of a point set surface in [*Amenta and Kil 05*] consists of two components:
  (i) a vector field $n(x)$
  and
  (ii) an energy function $e(y,n(x))$
- Point Set Surface is then defined as an extremal surface:

$$S = \{ x \mid x \in \arg\ local\ \min_{y \in \ell_{x,n(x)}} e(y,n(x)) \}$$

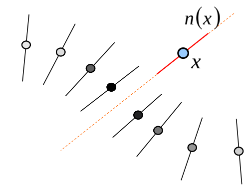


Projection Operator

Calculating projection direction (vector fields) and distance to surface (energy function).

(i) Vector fields:

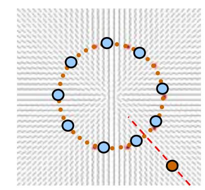
- **Local Neighborhood**



Compute a normal direction for each data point

Weighted average of closest directions [*Amenta and Kil 05*]

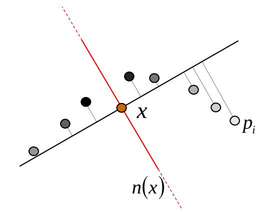
- **Gradient of Distance Field**



We apply Gaussian filter to smooth Euclidean distance field

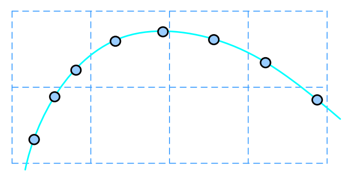
(ii) Energy function:

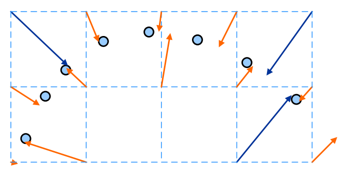
- **Moving Least Square**

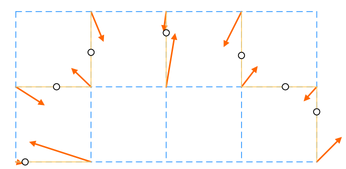


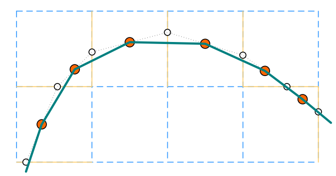
The energy term $e(x,n(x))$ measures how well the plane fits the data points

Process pipeline explained in 2D









- Input: Data point set
  Goal: Reconstruct the surface (blue) implied by this point set.
  Step 1: Lay a grid over the input point set.
- Step 2: At each grid intersection, calculate a vector (projection vector) toward the surface using projection operator.
- Step 3: Mark all grid edges with projection vectors at their two ends pointing in opposite directions. Approximate the intersection points between the surface and these edges based on the magnitudes of their projection vectors.
- Step 4: For each grid cell, calculate a grid center as the average of all intersection points. For each marked edge, connect the centers of its incident cells to make a piece of the surface.

Synthetic experiments:

- **Surface with boundary**

Original data

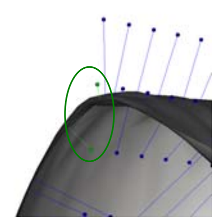
Reconstructed

- **Closed surface**

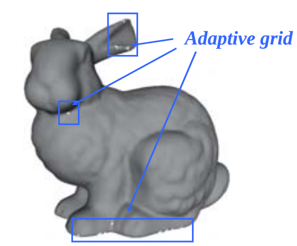
Original data

Reconstructed

Observations and Future Work



- At areas of high curvature, the vector field might change too fast, which leads to holes on the surface reconstructed.



- Adaptive grid, such as an octree, can be used to guarantee both speed and performance.