Computer graphics is not...
...only about modeling and image synthesis
How about acquiring geometry automatically?

Image courtesy: Stanford University
Digital Michelangelo Project

Image courtesy: Stanford University
Real-time 3-D acquisition

Image courtesy: Stanford University
Surface Reconstruction
Problem

Given a set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$ with $p_i \in \mathbb{R}^3$
Problem

Find a manifold surface $S \subset \mathbb{R}^3$ which approximates $\mathcal{P}$
Two approaches

Explicit

- Local surface connectivity estimation
- Point interpolation

Implicit

- Signed distance function estimation
- Mesh approximation
Two approaches

Explicit

– Ball pivoting algorithm
– Delaunay triangulation
– Alpha shapes
– ...

Implicit

– Distance from tangent planes
– SDF estimation via RBF
– ...

– Image space triangulation
Implicit surface reconstruction

Given a set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$ with $p_i \in \mathbb{R}^3$

Find a manifold surface $S \subset \mathbb{R}^3$ which approximates $\mathcal{P}$

where $S = \{x | d(x) = 0\}$ with $d(x)$ a signed distance function

Image courtesy: MPI Saarbrücken
Data flow

Point cloud

Signed distance function estimation

$$d(x)$$

Evaluation of distances on uniform grid

$$d(i), i = [i, j, k] \in \mathbb{Z}^3$$

Mesh extraction via marching cubes

Mesh
Implicit surface reconstruction methods

Mainly differ in their signed distance function
You will implement two popular methods

Hoppe’s distance from tangent planes

$C^0$ surface

Image courtesy: University of Washington
You will implement two popular methods

Signed distance function via triharmonic RBFs

$C^2$ surface

Image courtesy: University of Canterbury
Your new framework

reconstruct.cc
int main()
{
    load point cloud
    determine SDF
    evaluate SDF
    marching cubes
    write mesh
}

your code

ImplicitRBF → gmm
ImplicitHoppe
IsoEx
Input data

- ASCII file format
- List of point and normal coordinates

bunny-500.pts  sphere.pts
Exercise 2.1: 
Distance from tangent planes

\[ x \]

\[ x_i \]

\[ n_i \]
Find Closest point

\[ x \]

\[ n_i \]

\[ x_i \]
Distance to tangent plane

\[ d(x) = (x - x_i)^t n_i \]

Distance function is explicitly defined
→ no need to be estimated, evaluation is sufficient
Exercise 2.2: Signed distance function via RBFs

\[ d(x) = \sum_i w_i \phi(\|x - c_i\|) \]

- contribution weights
- radial basis function (kernel)
- centers

Must be estimated using constraints → Data interpolation problem
Estimate the signed distance function

\[ d(x) = \sum_i w_i \phi(\|x - c_i\|) \]

- Centers are chosen as constraint points
- Values constrained to points where distances are known
- Kernel is a triharmonic RBF: \( \phi(x) = x^3 \)
- Weights have to be determined
On- and off-surface constraints

\[ d(x) = \sum_i w_i \phi(\|x - c_i\|) \]

2n point constraints with known distances

\[ d(x_i + \epsilon n_i) = 1 \]

\[ d(x_i) = 0 \]
Linear system for weights

\[ d(x) = \sum_i w_i \phi(\|x - c_i\|) \]

\[ d(x_i) = 0 \]

\[ d(x_i + \epsilon n_i) = 1 \]

\[ \therefore \]

\[
\begin{bmatrix}
\phi(\|x_1 - x_1\|) & \ldots & \phi(\|x_1 - (x_n + \epsilon n_n)\|) \\
\vdots & \ddots & \vdots \\
\phi(\|(x_n + \epsilon n_n) - x_1\|) & \ldots & \phi(\|(x_n + \epsilon n_n) - (x_n + \epsilon n_n)\|)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_{2n}
\end{bmatrix}
=
\begin{bmatrix}
d_1 \\
\vdots \\
d_{2n}
\end{bmatrix}
\]

Solve system using the gmm library
Linear system for weights

\[ d(x) = \sum_i w_i \phi(\|x - c_i\|) \]

\[ d(x_i) = 0 \]

\[ d(x_i + \epsilon \mathbf{n}_i) = 1 \]

\[ \begin{bmatrix}
\phi(\|x_1 - x_1\|) & \cdots & \phi(\|x_1 - (x_n + \epsilon \mathbf{n}_n)\|) \\
\vdots & \ddots & \vdots \\
\phi(\|(x_n + \epsilon \mathbf{n}_n) - x_1\|) & \cdots & \phi(\|(x_n + \epsilon \mathbf{n}_n) - (x_n + \epsilon \mathbf{n}_n)\|)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_{2n}
\end{bmatrix}
=
\begin{bmatrix}
d_1 \\
\vdots \\
d_{2n}
\end{bmatrix} \]

distance \(d\) is fully defined
After a few seconds of computation you should obtain this
One more exercise...
Exercise 2.3 (optional): For the passionate

- Implement the Wendland RBF kernel

- Create more point data sets for implicit surface fitting

Image courtesy: Universität Göttingen
Further readings

• Surface reconstruction from unorganized points. [Hoppe et al. ‘92]

• A volumetric method for building complex models from range images [Curless and Levoy ‘96]

• Reconstruction and representation of 3D objects with radial basis functions [Carr et al. ‘01]