L5 – Implicit Surface Reconstruction

• Techniques to generate B-rep of surfaces with the help of implicit surfaces
  – Distance field
  – Radial Basis Function (RBF)
  – Multi-level Partition Unity (MPU) implicit
  – Poisson Reconstruction
  – Contouring methods for generating B-rep
    • Uniform sampling
    • Adaptive sampling
Implicit Functions & Implicit Surface

• In mathematics, an implicit function is a function in which the dependent variable has not been given "explicitly" in terms of the independent variable

• Implicit function based fitting (approximation or interpolation) is employed for surface reconstruction

• Advantages:
  – Compact mathematical representation
  – Easy topology change
  – Water-tight surface is always generated
Isoline and Isosurface

- An isoline of a function of two variables is a curve along which the function has a constant value.
- An isosurface is a 3D analog.
Distance Field Based Reconstruction

- Signed distance function: $f(p)$ from an arbitrary point $p$ in 3D to a known surface $M$ is the distance between $p$ and the closest point $z$ on $M$ multiplied by $\pm 1$
  - Sign depending on which side of the surface $p$ lies
  - Surface is defined at $f(p)=0$
- In reality $M$ is not known, but we can mimic this procedure using the oriented tangent planes
  - First, find the tangent plane $T_p(x_i)$ whose center $o_i$ is closest to $p$
  - The signed distance function is approximated by
    $$f(p) = \text{dist}_i(p) = (p - o_i) \cdot \hat{n}_i$$
    the distance between $p$ to its projection on the plane $T_p(x_i)$
Dist. Field

- **Piece-wise signed dist. field**
- **Main problems**
  - Influence of noises
  - Compatibility between tangent planes

\[
i \leftarrow \text{index of tangent plane whose center is closest to } p
\]

{ Compute \( z \) as the projection of \( p \) onto \( Tp(x_i) \) }
\[
z \leftarrow o_i - ((p - o_i) \cdot \hat{n}_i) \hat{n}_i
\]

\[
\text{if } d(z, X) < \rho + \delta \text{ then } f(p) \leftarrow (p - o_i) \cdot \hat{n}_i \quad \{= \pm ||p - z||\}
\]

else
\[
f(p) \leftarrow \text{undefined}
\]

endif
Mesh Generation from Distance Field

• A variation of **Marching Cubes algorithm**
  – To accurately estimate boundaries, the cube size should be set so that **edges** are of length **less** than \( \delta + \rho \)
  – Signed distance function \( f \) is evaluated only at points close to data
  – No intersection is reported within a cube if the signed distance function is undefined at any vertex of the cube, thereby giving rise to boundaries in the simplicial surface
  – As a result of MC, triangles with **poor** shape will be generated
  – How about the **intersection on edges**?
Problems of Distance Based Method

- Relies too much on the quality of input points
- The compatibility between neighboring tangent planes may have a problem
- Cannot guarantee to generate water-tight surface since there is some “undefined” region
RBF Based Surface Reconstruction

- Fitting an implicit function to the given points

\[
f(x_i, y_i, z_i) = 0, \quad i = 1, \ldots, n \quad \text{(on-surface points)},
\]
\[
f(x_i, y_i, z_i) = d_i \neq 0, \quad i = n + 1, \ldots, N \quad \text{(off-surface points}).
\]

- Mathematically, a good choice - radial basis function (RBF)

\[
s(x) = p(x) + \sum_{i=1}^{N} \lambda_i \phi(|x - x_i|)
\]

- \(p(x)\) is a low-degree polynomial
- The basis function \(\phi\) is a real-value function defined on the interval \([0, +\infty)\)

2D: \(\phi(r) = r^2 \log(r)\)

3D: \(\phi(r) = r^3\)  
\[
p(x) = c_1 + c_2x + c_3y + c_4z
\]
RBF Based Surface Reconstruction

• To ensure that the obtained surface has integrable second derivatives, the following side condition must be added

\[
\sum_{i=1}^{N} \lambda_i q(x_i) = 0, \text{ for all polynomials } q \text{ of degree at most } m.
\]

lead to

\[
\begin{pmatrix}
A & P \\
P^T & 0
\end{pmatrix}
\begin{pmatrix}
\lambda \\
c
\end{pmatrix}
= B
\begin{pmatrix}
\lambda \\
c
\end{pmatrix}
= 
\begin{pmatrix}
f \\
0
\end{pmatrix}
\]

E.g. \[
\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0
\]
when using 1st order polynomial in \( p(x) \)

\[A_{i,j} = \phi(|x_i - x_j|), \quad i, j = 1, \ldots, N,\]

\[P_{i,j} = p_j(x_i), \quad i = 1, \ldots, N, \quad j = 1, \ldots, \ell.\]

*The matrix \( B \) typically has poor conditioning as the number of data points \( N \) gets larger, and it is a dense matrix.
Solving RBF Based Reconstruction

• Direct solver does not work when $n>2,000$

• Fast Multi-pole Method (FMM) is employed
  – Fact: infinite precision is neither required nor expected
  – For the evaluation of an RBF, the approximations of choice are far- and near-field expansions
  – With the centers clustered in a hierarchical manner
  – far- and near-field expansions are used to generate an approximation to that part of the RBF due to the centers in a particular cluster

• Short Course at:
  www.math.nyu.edu/faculty/greengar/shortcourse_fmm.pdf
RBF Approximation of Noisy Data

- What if there are noises in data
- Consider this problem:
  \[ \min_{s \in BL^{(2)}(\mathbb{R}^3)} \rho \| s \|^2 + \frac{1}{N} \sum_{i=1}^{N} (s(x_i) - f_i)^2 \quad (\rho \geq 0) \]

- Solution can be obtained by
  \[
  \begin{pmatrix}
  A - 8N\pi \rho I \\
  P^T \\
  P
  \end{pmatrix}
  \begin{pmatrix}
  \lambda \\
  c
  \end{pmatrix}
  =
  \begin{pmatrix}
  f \\
  0
  \end{pmatrix}
  
  \text{Regularization Term}

- Another method: greedy reduction
  1. Choose a subset from the interpolation nodes \( x_i \) and fit an RBF only to these.
  2. Evaluate the residual, \( \varepsilon_i = f_i - s(x_i) \), at all nodes.
  3. If \( \max\{|\varepsilon_i|\} < \text{fitting accuracy} \) then stop.
  4. Else append new centers where \( \varepsilon_i \) is large.
  5. Re-fit RBF and goto 2.
A greedy algorithm iteratively fits an RBF to a point cloud.

Mesh Generation from Implicit Surface

- **Marching Cubes** algorithm or its variants
  - Resolutions fixed
  - Does not adapt to the curvature of surface
  - Topology homeomorphism is not guaranteed
  - Sharp features are damaged

- **Dual Contouring**
  - Could be adaptive
  - Quadrangular/triangular mesh
  - Singular vertices can be generated
  - Sharp features can be reconstructed
  - Less number of polygons
MC vs. DC
Partition of Unity

• Integrate \textit{locally} defined \textit{approximants} into a \textit{global} approx.

• A bounded domain $\Omega$ in 3D and a set of \textit{nonnegative compactly supported} functions: $\sum_i \varphi_i \equiv 1$ on $\Omega$

• Let us associate a local approximation set of functions $V_i$ with each sub-domain: $Q_i(x)$ (e.g., \textit{quadratic surface})

• Now an approximation of a function defined on $\Omega$ is

\[ f(x) \approx \sum_i \varphi_i(x) Q_i(x) \]

$Q_i \in V_i$

• Given a set of nonnegative compactly supported functions $\{w_i\}$, we have that to $\Omega \subset \bigcup_i \text{supp} \left( w_i \right)$, the $\{\varphi_i\}$ is defined by

\[ \varphi_i(x) = \frac{w_i(x)}{\sum_{j=1}^n w_j(x)} \]
Multi-level Partition of Unity Implicits

- For **approximation** purpose, using the quadratic B-spline function $b(t)$ as

$$w_i(x) = b\left(\frac{3|x-c_i|}{2R_i}\right)$$

where $c_i$ is the center, and $R_i$ is the support size.

- For **interpolation** purpose, using the inversed distance singular weights as

$$w_i(x) = \left[\frac{(R_i - |x-c_i|)_+}{R_i |x-c_i|}\right]^2,$$

where $(a)_+ = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$

- Octree based approximate: using $R$ as 0.75 of diagonal edge of each cell; or increase to enhance robustness.
MPU Implicits

• When different situations fit different local implicits
• Better results can be given

Apply MPU to RBF

- First, construct the hierarchy of pnts: \( \{ \mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^M = \mathcal{P} \} \)
- Then, starting from \( f^0(\mathbf{x}) = -1 \)

\[
f^k(\mathbf{x}) = f^{k-1}(\mathbf{x}) + o^k(\mathbf{x}) \quad (k = 1, 2, \ldots, M),
\]

where \( f^k(\mathbf{x}) = 0 \) interpolates \( \mathcal{P}^k \). An offsetting function \( o^k \)

\[
o^k(\mathbf{x}) = \sum_{\mathbf{p}_i^k \in \mathcal{P}^k} \left[ g_i^k(\mathbf{x}) + \lambda_i^k \right] \phi_{\sigma^k}(\| \mathbf{x} - \mathbf{p}_i^k \|).
\]

where local approximations \( g_i^k(\mathbf{x}) \) determined via least square fitting applied to \( \mathcal{P}^k \)

The shifting coefficients \( \lambda_i^k \) are found by solving the

\[
f^{k-1}(\mathbf{p}_i^k) + o^k(\mathbf{p}_i^k) = 0
\]
Compactly Supported RBF

\[ f(x) = \sum_{p_i \in \mathcal{P}} \psi_i(x) = \sum_{p_i \in \mathcal{P}} \left[ g_i(x) + \lambda_i \right] \phi_\sigma(||x - p_i||), \]

where \( \phi_\sigma(r) = \phi(r/\sigma) \), \( \phi(r) = (1 - r)^4 + (4r + 1) \) is Wendland’s compactly supported RBF [38], \( \sigma \) is its support size, and \( g_i(x) \) and \( \lambda_i \) are unknown functions and coefficients.
CSRBF Reconstruction

Left: a surface and feature points (ridge and ravine points) detected on it. Middle: only the feature points are kept. Right: surface reconstruction from the feature points only.

Interpolation of irregularly sampled data (73K points, 38 sec.).
Poisson Based Surface Reconstruction

- Compute a 3D indicator function: 1 for inside & 0 as outside
  - **Key insight:** there is an integral relationship between oriented points sampled from a model and its indicator function
  - Specifically, the gradient of the indicator function is a vector field that is zero almost everywhere (since the indicator function is constant), except the points near the surface, where it is equal to the inward surface normal

* The oriented point samples can be viewed as samples of the gradient of the model's indicator function
Poisson Reconstruction (cont.)

- The problem is converted to find the scalar function whose gradient best approximates a vector field defined by the samples – i.e. \( \min_\chi \| \nabla \chi - \vec{V} \| \)

- If we apply the divergence operator, this variational problem transforms into a standard Poisson problem:
  - Compute the scalar function whose Laplacian (divergence of gradient) \textbf{equals} the divergence of the vector field
    \[
    \Delta \chi \equiv \nabla \cdot \nabla \chi = \nabla \cdot \vec{V}
    \]

- It is a global solution but can still admit a hierarchy of locally supported functions, therefore its solution reduced to a well-conditioned sparse linear system
Numerical Computation

- Surface is not known, how can we evaluate the integral?
- Could we get an approximation from the input sample points?
- Using point set $S$ to partition surface into distinct patches $\mathcal{P}_s \subset \partial M$
- We can approximate the integral over a patch by the value at point sample $s.p$, scaled by the area of the patch

\[ \nabla (\chi_M \ast \tilde{F})(q) = \sum_{s \in S} \int_{\mathcal{P}_s} \tilde{F}_p(q) \tilde{N}_{\partial M}(p) dp \]

\[ \approx \sum_{s \in S} |\mathcal{P}_s| \tilde{F}_{s.p}(q) s.\tilde{N} = \tilde{V}(q) \]
Numerical Scheme for Computation

• Requirements on the filter:
  – should be sufficiently narrow so that do not over-smooth the data
  – should be wide enough so that the integral over a patch is well approximated by the value at $s.p$ scaled by the patch area

• Candidate: a Gaussian with variance being on the order of the sampling resolution

• Adaptive computation structure (in Octree)
  – Using the position of sample points to define the octree
  – Associate a function $F_o$ to each node of the tree

$$F_o(q) \equiv F \left( \frac{q - o.c}{o.w} \right) \frac{1}{o.w^3}.$$  

where $o.c$ and $o.w$ are the center and width of node $o.$
Reconstruction Result

• Not sensitive to noises
• Can fill holes effectively
• Preserve normals on samples
• Can be extended to run
  – Out-of-core
  – In parallel and on GPU
Closed-Form Formulation of HRBF-Based Surface Reconstruction

Input scenario with 922k points

Reconstruction on CPU within 5.5 sec. resulting in 313k triangles

- 17.9x faster than the state-of-the-art *Float Scaling Surface Reconstruction* (FSSR)