Surface Reconstruction with Radial Basis Functions

Pan Rongjiang

2007.10
Outline

- Surface from scattered points
- Implicit Surface Modeling with RBFs
- Multi-level Surface Reconstruction with RBFs
- A two-level surface fitting approach
- Future directions
1. Surface from scattered points

• Difficulties
  – large (millions of points)
  – irregular
  – noisy
  – incomplete
  – sharp features
  – redundancy

53044 points
(Courtesy Boštjan Pivec)
Reconstruction techniques

• Combinatorial structures based methods
  – Delaunay triangulations
  – Alpha shapes
  – Voronoi diagrams

• Implicit surface methods
  – Moving least squares (MLS)
  – Multi-level Partition of Unity (POU)
  – Radial Basis Function (RBF)
Implicit Surfaces

- Build a embedding or characteristic function.
- One connected isosurface is the implicit surface.
- Related to research in level sets.
- Not an parametric or polygonal surface representation.
Implicit Surface Methods

- impose no constraints on the topology
- repair incomplete data
- produce water-tight surfaces
- can be sampled with arbitrary precision to generate smooth models
- Some have analytic representations such as RBF which can be used to compute higher order derivatives
Implicit Function

- Goal: given N samples \((x_i, f_i)\), reconstruct a function \(s(x)\), such that \(s(x_i) = f_i\)
  - infinite solutions
- constraints: \(s(x)\) should be continuous over the entire domain since we want a smooth surface
  - RBF is a solution

(Courtesy Carr, 2001)
2. Implicit Surface Modeling with RBFs

• Given: a set of oriented points (constraints)
• Build a linear combination of radial basis functions (RBF) to create an embedding function.

\[ f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + P(\mathbf{x}) \]

- \( w_i \) - the weights of center (unknown)
- \( \phi(r) \) - the basis function
- \( P(x) \) - a low-degree polynomial (unknown)
- \( \|\mathbf{x}\| \) - the Euclidean norm
Finding an RBF Solution

• The weights and polynomial coefficients are unknowns

• We know N values of f(x_j): \( f_j, j = 1, \ldots, N \)

\[
f_j = w_1 \phi(\|x_j - x_1\|) + \ldots + w_N \phi(\|x_j - x_N\|) + c_0 + c_1 x_j + c_2 y_j + c_3 z_j
\]

• We also have 4 side conditions

\[
\sum_{i=1}^{N} w_i = \sum_{i=1}^{N} w_i x_i = \sum_{i=1}^{N} w_i y_i = \sum_{i=1}^{N} w_i z_i = 0
\]
The Linear System $Ax = b$

$$\sum w_i \phi_{ji} + P(x_j) = f_i$$

$$\sum w_i = 0$$

$$\sum w_i x_i = 0$$

$$\sum w_i y_i = 0$$

$$\sum w_i z_i = 0$$

$$\begin{bmatrix}
\phi_{11} & \cdots & \phi_{1N} & 1 & x_1 & y_1 & z_1 \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{N1} & \cdots & \phi_{NN} & 1 & x_N & y_N & z_N
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_N
\end{bmatrix}
= 
\begin{bmatrix}
f_1 \\
\vdots \\
f_N
\end{bmatrix}$$

where $\phi_{ji} = \phi(\|x_j - x_i\|)$
Basic Procedures

• Acquire N surface points with value 0
• Append off-surface points in order to avoid the trivial solution $w_i = 0$
• Constructing the system of equations
• Solve the linear system
• Evaluating the interpolating function in order to polygonize and render the surface.
Types of support

- **Global or Infinite** support:
  - not compact (no interval, goes to $\infty$)

- **Compact or Finite** support:
  - function value is zero outside of a certain interval
Globally supported RBFs

- not compact, goes to indefinite
- thin-plate spline radial basis function

$$\phi(r) = |r|^3$$

- Advantages
  - minimize bending energy of the embedding function
  - good extrapolation (hole-filling)
thin plate splines

- Interpolates across relatively large distances.
- Interpolating functions that minimize smoothness
  - metric (“bending energy”)

\[ E(f) = \int_{s \in \Omega} f^{2}_{xx}(s) + 2f^{2}_{xy}(s) + f^{2}_{yy}(s) \, ds \]

- Solved using radial basis functions (Duchon, 1978)
Problems of globally supported RBFs

• heavy computational cost
  – usually infeasible for more than a few thousand centres
  – Fast Multipole Methods make it feasible
    • mathematically complex
    • proprietary, commercial implementation (FarField Technology)

• undesirable property for shape modelling
  – a small change in even one constraint is felt throughout the entire resulting interpolated surface

• Spurious surface sheets
Approach - Change of RBF

- Shift to compactly supported RBF with high continuity.
- Can’t just use arbitrary functions
  - Smooth
  - Differentiability of resulting interpolation
  - Must produce positive definite matrix
- Gaussian:
  - Radially symmetric
  - Smooth result
  - Decays to zero rather than increasing
  - Effect of farther-away points is less rather than more
  - Matrix solution is better conditioned
Compactly supported RBFs (CSRBF)

• Can go one step further and use compact, locally-supported radial basis functions
  – function value is zero outside of a certain interval
  – Wendland's CSRBF

• Advantages
  – sparse matrix
  – fast algorithms that are easy to implement
  – practical for large models

\[
\phi(r) = \begin{cases} 
(1 - r)^4(4r + 1), & 0 \leq r \leq 1 \\
0, & \text{others}
\end{cases}
\]
Wendland's CSRBF

- Wendland (1995) has solved for minimum-degree compact functions that guarantee that the solution matrix is positive definite.

- can be scaled if needed

<table>
<thead>
<tr>
<th>$d = 1$</th>
<th>$(1 - r)_+^1$</th>
<th>$C^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(1 - r)_+^3(3r + 1)$</td>
<td>$C^2$</td>
</tr>
<tr>
<td></td>
<td>$(1 - r)_+^5(8r^2 + 5r + 1)$</td>
<td>$C^4$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$d = 3$</th>
<th>$(1 - r)_+^2$</th>
<th>$C^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(1 - r)_+^4(4r + 1)$</td>
<td>$C^2$</td>
</tr>
<tr>
<td></td>
<td>$(1 - r)_+^6(35r^2 + 18r + 3)$</td>
<td>$C^6$</td>
</tr>
<tr>
<td></td>
<td>$(1 - r)_+^8(32r^3 + 25r^2 + 8r + 1)$</td>
<td>$C^4$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$d = 5$</th>
<th>$(1 - r)_+^3$</th>
<th>$C^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(1 - r)_+^5(5r + 1)$</td>
<td>$C^2$</td>
</tr>
<tr>
<td></td>
<td>$(1 - r)_+^7(16r^2 + 7r + 1)$</td>
<td>$C^4$</td>
</tr>
</tbody>
</table>
Problems of CSRBFs

- It is only defined in a narrow band surrounding the shape
- It often yields surfaces with some unwanted artifacts and spurious zero level sets.
- It lacks of extrapolation across big holes
- The sign of $f(x)$ is not consistently negative inside the surface and positive outside the surface.
$\phi(r) = \left| r \right|^3$

<table>
<thead>
<tr>
<th></th>
<th>Thin-plate</th>
<th>Compact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation to build</td>
<td>$O(n^2)$</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>Computation to solve</td>
<td>$O(n^2)$</td>
<td>$O(n^{1.5})$</td>
</tr>
<tr>
<td>Storage to build/solve</td>
<td>$O(n^2)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Computation to evaluate</td>
<td>$O(n)$</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Effect of a single point</td>
<td>Global</td>
<td>Local</td>
</tr>
</tbody>
</table>

**Examples (Bunny model)**

- **2000 points**
  - 2:05 hrs vs. 5.93 sec
- **35947 points**
  - 39,434.4MB vs. 38.6MB

$\phi(r) = (1 - r)^3 (35r^2 + 18r + 3)$
Off-Surface Points

- Points inside and outside of surface
  - Project new centers along surface normals at points
  - Assign values: <0, inside; >0, outside
  - Projection distance has a large effect on smoothness
Point Normals

• Easy to get from polygonal meshes
• Difficult to get from point clouds
  – Can guess normal by fitting a plane to local neighborhood of points
  – Consistent normal Orientation
    • Propagate the orientation to neighbors by MST
    Or
    • Additional outward-pointing vector to determine orientation, e.g. range scanner position
  – For ambiguous cases, don’t generate off-surface point
data structures

• Octree
  – spatial subdivision

• k-d trees
  – find all points within some distance $r$ in log $n$ time.
3. Multi-level Surface Reconstruction with RBFs

- Center Redundant
  - using the whole data sets as RBF centers may cause the singularity of the linear system.

- Minor number of centers can also speed up:
  - interactive surface modeling
  - deformation
  - rendering
  - collision detection
Available methods

- coarse-to-fine greedy algorithm (Carr 2001)
  - solve and evaluate RBF iteratively
  - computationally expensive

- subsample points (Tobor 2004, Kitago 2006)
  - thinning point clouds is not an easy problem

- global iterative procedure (Patane 2006)
  - How to control the final approximation error remains unsolved
  - Sparsification ratio cannot increase too much

- Local Orthogonal Least Squares (Xia 2006)
  - cannot guarantee the global optimal surface representation
  - more centers are needed since Partition of Unity requires the overlapping area.
Our approach

• combines the benefits of local and global methods
• use the OLS algorithm to select centers locally.
  – Can speed up the global iterative approximate process
  – do not lose the accuracy
• apply global iterative process to further improve the compactness of the representation
  – enforce error constraints
Local OLS Criterion

- \( A = QR \)
  - \( Q \) is orthonormal
  - \( R \) is an upper triangular matrix

- The relative error is
  \[
  \frac{e^T e}{b^T b} = 1 - \sum_{i=0}^{K} \frac{(q_i^T b)^2}{b^T b}
  \]

- Error reduction ratio of the column \( q_i \) is
  \[
  r_i = \frac{(q_i^T b)^2}{(b^T b)}
  \]

- The ratio offers a simple and effective means to select the most significant centers
Global Iterative Least-squares Approximation

- Minimize the functional $F$ with respect to $f$
  
  \[
  F(f) = \sum_{i=1}^{M} (f(p_i) - b_i)^2 + 2\varepsilon \sum_{j=0}^{K} (w_j^2 + \eta)^{1/2}
  \]

  a smooth approximation of the $l_1$-norm of $w$

- It leads to the solution of a system of non-linear equations

  \[
  \left[ A^T A + \varepsilon \Delta(w) \right] w = A^T b \quad \Delta(w) = \begin{pmatrix}
  (w_0^2 + \eta)^{-1/2} & 0 \\
  0 & \ddots \\
  0 & 0 & (w_K^2 + \eta)^{-1/2}
  \end{pmatrix}
  \]
• the residual error of the least-square problem

\[ e^T e = (A w - b)^T (A w - b) = b^T b - b^T A w - \varepsilon w^T \Delta(w) w \]

• the number of zero components in \( w \) cannot decrease at successive iterations.

• If \( \frac{e^T e}{b^T b} < \) a pre-determined ratio \( \xi \), abandon the centers whose corresponding \( w_i \) is zero
evaluate the fitting accuracy

- Peak Signal to Noise Ratio (PSNR)
  - \textit{peak} is the diagonal length of the model’s bounding box
  - \( d \) is the average of algebraic sum of the Taubin distances

\[
PSNR[dB] = 20 \log_{10} \frac{\text{peak}}{d}
\]

\[
d = \frac{1}{N} \sum_{i=1}^{N} \frac{|f(p_i)|}{\|\nabla f(p_i)\|}
\]
Results

\[
\text{Ratio} = \frac{\#(\text{selected centers})}{\#(\text{input points})}
\]

<table>
<thead>
<tr>
<th>Ratio(%)</th>
<th>PSNR(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.0</td>
<td>62.2</td>
</tr>
<tr>
<td>32.0</td>
<td>61.8</td>
</tr>
<tr>
<td>28.0</td>
<td>61.6</td>
</tr>
<tr>
<td>20.0</td>
<td>59.0</td>
</tr>
<tr>
<td>15.3</td>
<td>51.8</td>
</tr>
<tr>
<td>11.3</td>
<td>48.8</td>
</tr>
</tbody>
</table>
Plots of the ratio and PSNR against relative error
Limitation

• The reconstructed surface has holes with a large approximation error
  – It is due to the use of compactly supported RBFs
4. A two-level surface fitting approach

- **CSRBF**
  - It is only as smooth as the thin-plate spline interpolation in a narrow band surrounding the shape
  - It often yields surfaces with some unwanted artifacts and spurious zero level sets.
  - The sign of $f(x)$ is not consistently negative inside the surface and positive outside the surface.
Problems of CSRBFs in Modeling

- It causes no serious problems in polygonization and rendering
- It is not suitable for collision detection and set operations between shapes.
- We want the sign of $f(x)$ is meaningful inside the bounding box
Available methods

• Reconstruction with Voronoi Centered RBFs (Samozino 2006)
  – Using centers selected among the vertices of the Voronoi diagram of the input data points
  – it needs a user-defined budget of centers

• Reconstruction with globally regularised CSRBFs (Walder 2006)
  – a complicated expression has to be evaluated
double regulariser(double r, double s1, double s2) {
    return
    (32* pow(3.14159265358979323846, 8)+ 6*s1*(-48+pow(r,5) + 
    240* pow(r,3) * pow(s1,2) + 360* pow(r,2) * pow(s1,3) + 210* r* pow(s1,4) + 
    45* pow(s1,5) + 80* pow(r,4) * s2 - 240* pow(r,2) * pow(s1,2) * s2 - 
    240*r* pow(s1,3) * s2 - 70* pow(s1,4) * s2 - 80* pow(r,2) * pow(s2,3) + 
    40* pow(s1,2) * pow(s2,3) - 60*r* pow(s2,4) - 16* pow(s2,5) - 
    96* pow(fabs(r - s1, 6) + 6* pow(fabs(-2*r + s1, 5) + pow(fabs(-2*r + 
    s1 - 2*s2), 5) + 16* pow(fabs(r - s1 - s2), 5) + 4* pow(fabs(2*r + s1 - s2), 5) - 2* pow(fabs(-2*r - 
    2*r + s1 + s2), 5) + 2* pow(fabs(2*r + s1 - s2), 5) - 16* pow(fabs(r - 
    s1 - s2), 5) - 4* pow(fabs(2*r - s1 + s2), 5) + 4* pow(fabs(-2*r + s1 + 
    s2), 5) - 16* pow(fabs(r - s1 + s2), 5) + 2* pow(fabs(-2*r + 2*r + s1 + 
    s2), 5) + pow(fabs(-2*r + s1 + 2*s2), 5)) + 6*s2*(-48+pow(r,5) + 
    80* pow(r,4) * s1 - 80* pow(r,2) * pow(s1,3) - 60*r* pow(s1,4) - 
    14* pow(s1,5) + 240* pow(r,3) * pow(s2,2) - 240* pow(r,2) * s1* pow(s2,2) + 
    40* pow(s1,3) * pow(s2,2) + 360* pow(r,2) * pow(s2,3) + 
    240*r* s1* pow(s2,3) + 210*r* pow(s2,4) - 70*s1* pow(s2,4) + 
    45* pow(s2,5) - 2* pow(fabs(-2*r + s1 - 2*s2), 5) + 2* pow(fabs(2*r + s1 - 2*s2), 5) - 
    96* pow(fabs(r - s2, 5) + 16* pow(fabs(r - s1 - s2), 5) + 4* pow(fabs(2*r + s1 - s2), 5) + 
    pow(fabs(-2*r + s1 + s2), 5) + 16* pow(fabs(r - s1 + s2), 5) - 
    4* pow(fabs(-2*r + 2*r + s1 + s2), 5) + 16* pow(fabs(-2*r + r + s1 + 
    s2), 5)) + 4*(-576* pow(r, 8) - 96* pow(r + s1, 6) + 6* pow(2*r + s1, 6) - 96* pow(r + 
    s2, 6) - 6* pow(2*r + s2, 6) - 16* pow(r + s1 + s2, 6) + 16* pow(r + 
    s2, 6) + 2* pow(r + s1 + s2, 6) - 16* pow(r + s1 + s2, 6) + 16* pow(r + 
    s2, 6) - 96* pow(r - s1, 6)* sign(r - s1) + 6* pow(-2*r + s1, 6)* sign(r - s1/2.0) - 
    96* pow(r - s2, 6)* sign(r - s2) - 16* pow(r - s1 + s2, 6)* sign(r - 
    s1 - s2) + pow(-2*r + s1 + 2*s2, 6) + sign(r - s1/2.0) - s2) + 
    16* pow(r + s1 - s2, 6)* sign(r - s1 - s2) - 4* pow(2*r + s1 - s2), 6)* sign(2*r - 
    s1 + s2) + pow(-2*r + s1 - 2*s2, 6) + sign(r - s1/2.0) + 
    15*s1*s2*(-8* pow(r + s1 + s2, 4) - 2* pow(2*r + s1 + s2, 4) + pow(2*r + 
    s1 + s2, 4) + pow(2*r + s1 + s2, 4) + pow(2*r + s1 + s2, 4) - 8* pow(-r + s1 + 
    s2, 4) + pow(r - s1 - s2, 4) + pow(-2*r + s1 + 2*s2, 4) + sign(r - s1/2.0) - 
    s2) - pow(2*r + s1 - 2*s2, 4)* sign(r + s1/2.0) - 2* 8* pow(r + s1 - 
    s2, 4)* sign(r + s1 - s2) + 2* pow(2*r + s1 - s2, 4)* sign(r + s1 - s2) + 
    pow(2*r - s1 - s2, 4) - 2* pow(2*r + s1 - 2*s2, 4) + sign(r - s1 - s2, 4) + sign(r - s1 - s2, 4) + sign(r - s1 - s2, 4) + sign(r - s1 - s2, 4)) / (5.* pow(s1,5)* pow(s2,5));
}

regulariser

(Courtesy Walder, 2006)
A two-level fitting approach

- **Coarse scale approximation**
  - set centers of basis functions by a non-uniform grid
  - compute their signed distances to the underlying surface

- **Fine Scale Interpolation**
  - add off-surface points according to the residual errors
  - fitting to the residual errors on the surface points and additional off-surface points

- **The final function = Coarse + Fine**
  - It is a good approximation of the signed distance field to the surface inside the bounding box.
4.1 Coarse Scale approximation

- The support size $\sigma$ is estimated by an octree-based data structure
  - each leaf cell containing no more than eight points
  - delete the leaf cells containing no points
  - $\sigma = 0.75 \times$ average diagonal length of the leaf cells.

- enlarge the axis-aligned bounding box (AABB)

$$(x_{\min} - 2\sigma, x_{\max} + 2\sigma) \times (y_{\min} - 2\sigma, y_{\max} + 2\sigma) \times (z_{\min} - 2\sigma, z_{\max} + 2\sigma)$$
Centers by a non uniform grid

- put a grid of spacing $4\sigma$ on the the enlarged AABB
- For each grid cell that contains sample points, subdivide it into eight subcells of spacing $2\sigma$.
- If the subcell contains points, decompose it into eight smaller subcells of width $\sigma$.
- The centers of cells and subcells are selected as the centers of basis functions.
Estimate function values on centers

• for each center \( \mathbf{c} \),
  – find the nearest input point \( \mathbf{p} \)
  – fit a local quadratic approximation at \( \mathbf{p} \) to \( \mathbf{p} \)'s \( k \) nearest neighborhoods (we choose \( k=20 \))
    \[
g(u, v, w) = w - (Au^2 + 2Bu + Cv^2 + Du + Ev + F)
    \]
  – use Taubin’s first-order approximation of the signed distance
    \[
d_i = \frac{g(u_i, v_i, w_i)}{\|\nabla g(u_i, v_i, w_i)\|}
    \]
• Since $d_i$ is only meaningful in the nearby of the local quadratic fitting function, we use:

$$d_i = (c_i - p_j)^T n_j \quad \text{if} \quad \|c_i - p_j\| < |d_i|$$

• the support size:
  – three fourth of the diagonal length of the coarsest cell

$$\sigma_1 = 3\sqrt{3}\sigma$$

• coarse scale approximation $f^1$

$$f^1(x) = \sum_{j=1}^{K} w_j \phi_{\sigma_1}(\|x - c_j\|) + P(x)$$
4.2 Fine Scale Interpolation

• If there is no center in some thin parts of the point set surfaces, the approximate signed distance field is not correct in these regions.
Observation

• The thin parts of the model, which cannot be correctly approximated by coarse scale centers, must be included in a subcell of size $\sigma$.

• Residuals of surface points are usually large in these thin parts than in other parts.

• The thin parts need more constraints.
Add off-surface points

• Calculate the residual at each input point
  \[ r_i = f_i - f^1(p_i) \]

• If \[ |r_i| > \epsilon \], add an off-surface points along its normal
  – implemented in a bisection algorithm
  – in our implementation, \( \epsilon = \frac{1}{N} \sum_{i=1}^{N} |r_i| \)
Bisection algorithm

\[
\begin{align*}
distHigh & \leftarrow 0.5 \times \sigma \\
distLow & \leftarrow 0.0 \\
\text{if } (r_i > \varepsilon) \text{ then} \\
& \quad q \leftarrow p_i - distHigh \times n_i \\
\text{else} \\
& \quad q \leftarrow p_i + distHigh \times n_i \\
\text{end if} \\
\text{if } (p_i \in \mathcal{P} \text{ is not the nearest point of } q) \text{ then} \\
& \quad \text{while } (distHigh > distLow) \text{ do} \\
& \quad \quad \text{distMiddle} \leftarrow (\text{distHigh} + \text{distLow})/2 \\
& \quad \quad \text{if } (r_i > \varepsilon) \text{ then} \\
& \quad \quad \quad q \leftarrow p_i - \text{distMiddle} \times n_i \\
& \quad \quad \text{else} \\
& \quad \quad \quad q \leftarrow p_i + \text{distMiddle} \times n_i \\
& \quad \quad \text{end if} \\
& \quad \text{if } (p_i \in \mathcal{P} \text{ is the nearest point of } q) \text{ then} \\
& \quad \quad \text{distLow} \leftarrow \text{distMiddle} \\
& \quad \text{else} \\
& \quad \quad \text{distHigh} \leftarrow \text{distMiddle} \\
& \quad \text{end if} \\
& \quad \text{end while} \\
& \text{end if} \\
& \text{estimate the signed distance } dist \text{ from } q \text{ to } p_i
\end{align*}
\]
The final RBF function

• The final RBF function is

\[ f(x) = f^1(x) + f^2(x) \]

• To evaluate the fitting accuracy
  – Peak Signal to Noise Ratio (PSNR)
    • \textit{peak} is the diagonal length of the model’s bounding box
    • \textit{d} is the average of algebraic sum of the Taubin distances

\[ PSNR[dB] = 20 \log_{10} \frac{\text{peak}}{d} \]
### Results for five models

(1.80 GHz Pentiumn 4, 1GB )

<table>
<thead>
<tr>
<th>Model</th>
<th>#Points</th>
<th>#Centers 1</th>
<th>#Centers 2</th>
<th>PSNR (dB)</th>
<th>Tcoarse (s)</th>
<th>Tfine (s)</th>
<th>Ttotal (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2Torus</td>
<td>4352</td>
<td>884</td>
<td>6231</td>
<td>169.09</td>
<td>0.98</td>
<td>2.58</td>
<td>3.56</td>
</tr>
<tr>
<td>Knot</td>
<td>28659</td>
<td>10640</td>
<td>39802</td>
<td>123.97</td>
<td>11.84</td>
<td>44.13</td>
<td>55.97</td>
</tr>
<tr>
<td>Bunny</td>
<td>34835</td>
<td>14903</td>
<td>48842</td>
<td>189.79</td>
<td>17.86</td>
<td>16.91</td>
<td>34.77</td>
</tr>
<tr>
<td>Hand</td>
<td>36616</td>
<td>11064</td>
<td>46258</td>
<td>188.25</td>
<td>14.54</td>
<td>54.06</td>
<td>68.60</td>
</tr>
<tr>
<td>Armadillo</td>
<td>165954</td>
<td>77595</td>
<td>228626</td>
<td>139.25</td>
<td>110.14</td>
<td>460.33</td>
<td>570.50</td>
</tr>
</tbody>
</table>
4352 points
3.56 (s)

28659 points
55.97 (s)

coarse scale  fine scale  planar slice
34835 points
34.77 (s)

36616 points
68.60 (s)

coarse scale  fine scale  planar slice
165954 points, 570.50s

course scale  

fine scale  

planar slice
hole filling
Set operations - Union

\[ U \cup \text{object} = \text{combined object} \]

\[ U \cup \text{object} = \text{combined object} \]
Set operations - Intersection

\[ \text{rabbit} \cap \text{ring} = \text{shaped object} \]

\[ \text{rabbit} \cap \text{sphere} = \text{shape} \]
Set operations - Difference
Future Work

• Better ways to deal with extracting desired isosurface and not the inner/outer hulls
  – Ray tracing is possible, zero crossings with high gradient magnitude
• More accurate model simplification
• Reconstruction without off-surface points
• Reconstruction without normals
• Choice of center position and different supports