Wavelets for Surface Reconstruction

Josiah Manson
Guergana Petrova
Scott Schaefer
Convert Points to an Indicator Function
Data Acquisition
## Properties of Wavelets

<table>
<thead>
<tr>
<th></th>
<th>Fourier Series</th>
<th>Wavelets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Represents</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>all functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Locality</strong></td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Smoothness</strong></td>
<td>✓</td>
<td>Depends on wavelet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Wavelet Bases

Haar

\( \varphi(x) \)

\( \psi(x) \)

D4

\( \varphi(x) \)

\( \psi(x) \)
Example of Function using Wavelets

\[ f(x) = \sum_k c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]
Example of Function using Wavelets

\[ f(x) = \sum_k c_k \phi(x-k) \]
Example of Function using Wavelets

\[ f(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j=0,k} c_{j,k} \psi(2^j x - k) \]
Example of Function using Wavelets

$$f(x) = \sum_{k} c_{k} \varphi(x-k) + \sum_{j \in \{0,1\}, k} c_{j,k} \psi(2^{j} x - k)$$
Example of Function using Wavelets

\[ f(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j=0,1,2, k} c_{j,k} \psi(2^j x - k) \]
Strategy

- Estimate wavelet coefficients of indicator function
- Use only local combination of samples to find coefficients
Computing the Indicator Function

[2005 Kazhdan]

\[ \chi(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]
Computing the Indicator Function

\[ \chi(x) = \sum_k c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[ c_{j,k} = \int_{R} \chi(x) \psi(2^j x - k) dx \]

[Kazhdan 2005]
Computing the Indicator Function

\[ \chi(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[ c_{j,k} = \int_{R} \chi(x) \psi(2^j x - k) dx = \int_{M} \psi(2^j x - k) dx \]
Computing the Indicator Function

\[ \chi(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[ c_{j,k} = \int_{R} \chi(x) \psi(2^j x - k) dx = \int_{M} \psi(2^j x - k) dx \]

Divergence Theorem

\[ \int_{M} \nabla \cdot \vec{F}(x) dx = \int_{\partial M} \vec{F}(p) \cdot \vec{n}(p) d\sigma \]
Computing the Indicator Function

\[ \chi(x) = \sum_k c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[ c_{j,k} = \int_R \chi(x) \psi(2^j x - k) dx = \int_M \psi(2^j x - k) dx \]

Divergence Theorem

\[ \int_M \nabla \cdot \vec{F}(x) dx = \int_{p \in \partial M} \vec{F}(p) \cdot \vec{n}(p) d\sigma \]
Computing the Indicator Function

\[ \chi(x) = \sum_{k} c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[
c_{j,k} = \int_{R} \chi(x) \psi(2^j x - k) dx = \int_{M} \psi(2^j x - k) dx
\]

\[
= \int_{p \in \partial M} \tilde{F}_{j,k}(p) \cdot \tilde{n}(p) d\sigma
\]

[Kazhdan 2005]
Computing the Indicator Function

[Kazhdan 2005]

\[ \chi(x) = \sum_{k} c_{k} \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

\[ c_{j,k} = \int_{R} \chi(x) \psi(2^j x - k) dx = \int_{M} \psi(2^j x - k) dx \]

\[ = \int_{p \in \partial M} \vec{F}_{j,k}(p) \cdot \vec{n}(p) d\sigma \]

\[ \approx \sum_{i} \vec{F}_{j,k}(p_i) \cdot \vec{n}_i d\sigma_i \]
Finding $\vec{F}(x)$

$$\nabla \cdot \vec{F}(x) = \psi(2^j x - k)$$
Finding $\vec{F}(x)$

$$\nabla \cdot F(x) = \frac{d}{dx} F(x) = \psi(2^j x - k)$$
Finding $\vec{F}(x)$

$$\nabla \cdot F(x) = \frac{d}{dx} F(x) = \psi(2^j x - k)$$

$$F(x) = \int_{-\infty}^{x} \psi(2^j s - k) ds$$
Extracting the surface

\[ \chi(x) = \sum_k c_k \varphi(x - k) + \sum_{j,k} c_{j,k} \psi(2^j x - k) \]

Coefficients \rightarrow Indicator function \rightarrow Dual marching cubes \rightarrow Surface
Smoothing the Indicator Function

Haar unsmoothed
Smoothing the Indicator Function

Haar unsmoothed

Haar smoothed
Comparison of Wavelet Bases

Haar

D4
Advantages of Wavelets

• Coefficients calculated only near surface
  – Fast
  – Low memory

• Multi-resolution representation

• Out of core calculation is possible
Results

Michelangelo’s Barbuto

329 million points (7.4 GB of data), 329MB memory, 112 minutes
Results

Michelangelo’s Awakening
381 million points (8.5 GB), 573MB memory, 81 minutes
Produced 590 million polygons
Results

Michelangelo’s Atlas
410 million points (9.15 GB), 1188MB memory, 98 minutes
Produced 642 million polygons
Results

Michelangelo’s Atlas
410 million points (9.15 GB), 1188MB memory, 98 minutes
Produced 642 million polygons
Robustness to Noise in Normals

0°  30°  60°  90°
Comparison of Methods

- **MPU**
  - Time: 551 sec
  - Memory: 750 MB

- **Poisson**
  - Time: 289 sec
  - Memory: 57 MB

- **Haar**
  - Time: 17 sec
  - Memory: 13 MB

- **D4**
  - Time: 82 sec
  - Memory: 43 MB
Conclusions

- Wavelets provide trade-off between speed/quality
- Works with all orthogonal wavelets
- Guarantees closed, manifold surface
- Out of core