Non-Iterative, Feature-Preserving Mesh Smoothing

Thouis R. Jones (MIT), Frédo Durand (MIT), Mathieu Desbrun (USC)

thouis@graphics.csail.mit.edu, fredo@graphics.csail.mit.edu, desbrun@usc.edu
Why Smooth?

3D scanners are noisy...
Why Smooth?

3D scanners are noisy...
and have dropouts...

Jones, Durand, Desbrun
Why Smooth?

3D scanners are noisy... and have dropouts... and usually require multiple scans.

Jones, Durand, Desbrun
Goals

Fast smoothing of meshes

Robust

- Geometrically: preserve features
- Topologically: no connectivity information

Simple to implement
Goals

Fast smoothing of meshes polygon soups

Robust

- Geometrically: preserve features
- Topologically: no connectivity information

Simple to implement
Previous Work on Smoothing

Fast Mesh Smoothing
  • Taubin 1995; Desbrun et al. 1999

Feature Preserving
  • Clarenz et al. 2000; Desbrun et al. 2000; Meyer et al. 2002; Zhang and Fiume 2002; Bajaj and Xu 2003

Diffusion on Normal Field
  • Taubin 2001; Belyaev and Ohtake 2001; Ohtake et al. 2002; Tasdizen et al. 2002

Wiener Filtering of Meshes
  • Peng et al. 2001; Alexa 2002; Pauly and Gross 2001 (points)
Approach

We cast feature-preserving filtering as a robust estimation problem on vertex positions.

Extend Bilateral Filter to 3D.
- Smith and Brady 1997; Tomasi and Manduchi 1998

Use first-order predictors based on facets of model.
Single pass.
Non-Robust Estimation

Least Squares Error Norm

Outliers have unlimited influence on estimate.

Jones, Durand, Desbrun
Outliers have bounded influence on estimate.

Robust Estimation

Robust Error Norm
Gaussian Filter (Non-robust)

\[ I'_s = \sum_p I(p) f(s - p) \]

Jones, Durand, Desbrun
Bilateral Filter (Robust)

\[
I'_s = \frac{1}{k_s} \sum_p \left( I(p) \cdot f(s-p) \cdot g(I_s - I_p) \right)
\]
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{k_s} \sum_p I(p) \cdot f(s - p) \cdot g(I_s - I_p) \]

Jones, Durand, Desbrun
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{k_s} \sum_p I(p) f(s - p) g(I_s - I_p) \]

Jones, Durand, Desbrun
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{\kappa_s} \sum_{p} I(p) f(s - p) g(I_s - I_p) \]

\[ I \\ f \\ g \\ fg \\ I' \]
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{\kappa_s} \sum_p I(p) f(s - p) g(I_s - I_p) \]
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{k_s} \sum_p I(p) \cdot f(s - p) \cdot g(I_s - I_p) \]
Bilateral Filter (Robust)

\[ I'_s = \frac{1}{k_s} \sum_p \frac{I(p)}{f(s-p)} g(I_s - I_p) \]

\[ k_s = \sum_p f(s-p) g(I_s - I_p) \]
Bilateral Filter

Left: Jones and Jones 2003
Right: Bilaterally filtered.

Jones, Durand, Desbrun
Extending the Bilateral Filter to Meshes

How to separate location and signal in a 3D model?
• Forming local frames requires a connected mesh.

Instead, use first-order predictors based on facets:

No connectivity required between facets.
Bilateral Filter for Meshes

Estimate $p'$, the new position for a vertex $p$

$$p' = \frac{1}{k(p)} \sum_{q \in S} \Pi_q(p) \cdot f(||c_q - p||) \cdot g(||\Pi_q(p) - p||) \cdot a_q$$
Bilateral Filter for Meshes

Estimate $p'$, the new position for a vertex $p$

$$p' = \frac{1}{k(p)} \sum_{q \in S} \begin{cases} 
\text{prediction} \left\{ \Pi_q(p) \right\} \\
\text{spatial} \left\{ f(||c_q - p||) \right\} \\
\text{influence} \left\{ g(||\Pi_q(p) - p||) \right\} \\
\text{area} \left\{ a_q \right\}
\end{cases}$$

Jones, Durand, Desbrun
Bilateral Filter for Meshes

Estimate $p'$, the new position for a vertex $p$

$$p' = \frac{1}{k(p)} \sum_{q \in S} \Pi_q(p) \cdot f(||c_q - p||) \cdot g(||\Pi_q(p) - p||) \cdot a_q$$
Estimate $p'$, the new position for a vertex $p$

\[ p' = \frac{1}{k(p)} \sum_{q \in S} \left( \Pi_q(p) \cdot f(||c_q - p||) \right) \cdot g(||\Pi_q(p) - p||) \cdot a_q \]
Why we expect it to work

Predictions across corners are "outliers".
Dealing with Noise

Noise has a nonlinear effect on predictions.
Dealing with Noise

Noise has a nonlinear effect on predictions. We must *mollify* (pre-smooth) normals.
Dealing with Noise

Noise has a nonlinear effect on predictions. We must *mollify* (pre-smooth) normals.
Dealing with Noise

Noise has a nonlinear effect on predictions. We must *mollify* (pre-smooth) normals.
Dealing with Noise

Noise has a nonlinear effect on predictions. We must *mollify* (pre-smooth) normals.
Implementation

3K vertices / second (typical), 1.4 GHz Athlon.
Gaussians for $f$ and $g$.

Optimizations
  - Cutoff at twice spatial filter radius.
  - Binning for spatially coherent computation.

Data and non-optimized code available online.
Results - Smoothing

Original  Desbrun 1999  Our result

Jones, Durand, Desbrun
Results - Effect of $g$

Original

Without $g$

Our result

Jones, Durand, Desbrun
Results - Effect of Mollification

Original         Without mollification         Our result

Jones, Durand, Desbrun
Results - Connectivity

50% Original  Smoothed  All predictors

Jones, Durand, Desbrun
Results - Varying width of $f$ and $g$

Original

Narrow spatial and influence

Jones, Durand, Desbrun
Results - Varying width of $f$ and $g$

Original Narrow spatial and wide influence

Jones, Durand, Desbrun
Results - Varying width of $f$ and $g$

Original  Wide spatial and influence

Jones, Durand, Desbrun
Normalization factor $k$ as "Confidence"

Normalization term $k(p)$ is sum of weights, and is a measure of confidence in the estimation at $p$.

$$p' = \frac{1}{k(p)} \sum_{q \in S} \Pi_q(p) \ f(||c_q - p||) \ g(||\Pi_q(p) - p||) \ a_q$$

$$k(p) = \sum_{q \in S} f(||c_q - p||) \ g(||\Pi_q(p) - p||) \ a_q$$
Results - $k$ as Confidence
Results - $k$ as Confidence

Jones, Durand, Desbrun
Results - vs Wiener Filtering
Results - vs Wiener Filtering (Low Noise)

Peng et al. 2001

Our result

Jones, Durand, Desbrun
Results - vs Wiener Filtering (High Noise)

Peng et al. 2001

Our result

Jones, Durand, Desbrun
Results - vs Anisotropic Diffusion

Original

Jones, Durand, Desbrun
Results - vs Anisotropic Diffusion

Clarenz et al. 2000

Our result
Similar Methods

Bilateral Mesh Denoising, Fleishman et al. 2003 (next talk)
- Iterative
- Local frame
- No mollification
- Different predictor

Trilateral Filter, Cloudhury and Tumblin 2003 (EGSR)
- Images and Meshes
- Mollify normals, then vertices
- Different predictor
Future Work

Extend to other types of data (point models, volume data).

Using $k$ to steer further processing.

Iterative application.
Conclusions

Fast, feature preserving filter.
Simple to implement.
Applicable to polygon soups.

Take-home message:

- Robust estimation for smoothing.
- Points across features are outliers.
- First-order predictors remove connectivity requirements.
Acknowledgements

SIGGRAPH reviewers, Caltech SigDraft and MIT pre-reviewers.

Udo Diewald, Martin Rumpf, Jianbo Peng, Denis Zorin, and Jean-Yves Bouguet, and Stanford 3D Scanning Repository for models.

Peter Shirley and Michael Cohen for comments on this presentation.

We would like to thank the NSF (CCR-0133983, DMS-0221666, DMS-0221669, EEC-9529152, EIA-9802220).