Normal Improvement for Point Rendering

Thouis R. Jones, Frédo Durand, and Matthias Zwicker
Massachusetts Institute of Technology

A bilateral filter-based method for denoising normals for point models treats the filter as a spatial deformation and updates normals iteratively.

Models created from 3D scanners are becoming more prevalent as the demand for realistic geometry grows and scanners become more common. Unfortunately, scanned models are invariably noisy. This noise corrupts both samples' positions and normals. Many methods have been proposed for denoising models' geometry, whether represented as triangle meshes or as points. Less effort has been spent on improving normals, except as a step toward or byproduct of smoothing geometry, and even though noise in normals affects rendering quality more than noise in positions. We believe there are benefits to considering normal improvement on its own.

Our proposed method for improving normals is derived from a feature-preserving geometry filter. Many such filters are available, most operating on models represented as triangles meshes.

We argue that for point rendering, removing noise from normals is more important than removing noise from geometry, because normals have a greater impact on the model's perceived quality. Nonlinearity in lighting calculations causes even low levels of noise in normal directions to be quite noticeable, while the level of positional noise in scanned models is seldom enough to cause visible occlusion errors. Normal filtering has been explored by others, but generally as a step toward smoothing of geometry,[1,2] and not as an end in itself.

Our filter does not modify sample positions during smoothing. We could attempt to simultaneously smooth normals and positions, but similarly to lighting calculations, noise in normals has a nonlinear effect on the estimates of smooth positions. This leads to poor smoothing in flat areas and over smoothing near features.[3] We avoid these difficulties by concentrating on normals alone.

Two approaches for smoothing point models have been proposed. Point set surfaces estimate smoothed normals and geometry by least-squares fitting to locally weighted neighborhoods.[4] The spectral processing method creates a local height field, which is then filtered and resampled.[5] The former method is not feature preserving, while the latter requires resampling to a regular grid, which can degrade features. Our method is novel in that it preserves features and doesn't require resampling.

3D bilateral filter

The bilateral filter was originally proposed in image processing,[6,7] but there have been three recent extensions to 3D shapes.[2,3,8] We use the 3D bilateral filter proposed by Jones, Durand, and Desbrun[3] because of its straightforward extension to models represented as points with normals, such as surfel models.[9]

The 3D bilateral filter applied to a model predicts a new position for every point as a weighted combination of predictions from nearby points in the model, based on their positions and normals. It is of the form

$$s' = \frac{1}{k(s)} \sum_{p \in S} \Pi_p(s) f\left(\left\|s-p\right\|\right) g\left(\left\|\Pi_p(s)-s\right\|\right) \tag{1}$$

where S is the set of all points in the model, k is a normalizing factor (sum of the weights),

$$k(s) = \sum_{p \in S} f\left(\left\|s-p\right\|\right) g\left(\left\|\Pi_p(s)-s\right\|\right)$$

and $\Pi_p(s)$ is the linear prediction for s given the information at point p, $\Pi_p(s) = s + (p - s) \cdot n_p\, n_p$, where n_p is the normal at p. This is simply the projection of s onto the plane through p with normal n_p.

We refer to the two functions f and g as the spatial weight and influence weight functions, respectively. They are both positive, monotonic, decreasing functions. The first controls how large a neighborhood of points is used to estimate s'. The second causes predictions $\Pi_p(s)$ that are far from the original position s to have less effect on the estimate for s'. In other words, g rejects outliers and gives the 3D bilateral filter its feature-preserving behavior. We use Gaussians of width σ_f and σ_g in our work. The feature-preserving nature of the bilateral filter and its relation to the choice for f and

Analysis of the Transposed Adjoint of the Filter

The effectiveness of our method can be understood by analysis of an ideal case in 2D. Suppose we have an infinite planar surface defined by $y = 0$ with constant normal $[0\ \ 1]^T$, and are adjusting the normal for a point $s=[s_1\ \ s_2]^T$ near the surface. As an approximation, and since we are near the surface, we can neglect the influence of s itself on the prediction for its smoothed position s'. In this case, it's not difficult to see that s' will always be the perpendicular projection of s onto $y = 0$, that is, $s' = F(s) = [s_1\ \ 0]^T$. Thus, the Jacobian of F will be

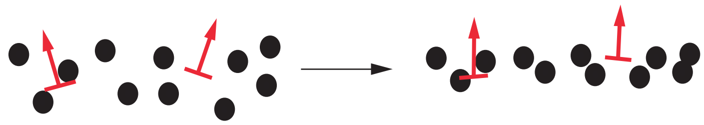
$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

with transposed adjoint

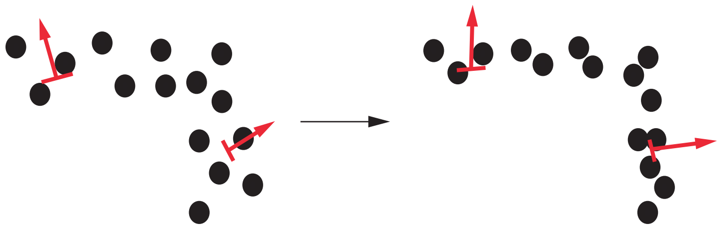
$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

In the more realistic case, in which the surface's sample positions and normals are noisy, the s' might not move completely to the $y = 0$ plane, or it could drift tangentially relative to s. However, if $s' = F(s) \approx [s_1\ \ \varepsilon s_2]^T$ with $\varepsilon < 1$ (as will be true for reasonable levels of noise), then the transpose adjoint will be near

$\begin{bmatrix} \varepsilon & 0 \\ 0 & 1 \end{bmatrix}$, and after sufficient iterations, n_s will be close to $[0\ \ 1]^T$.



1 How the filter affects normals near a flat region of the surface.



2 How the filter affects normals near a corner.

g is discussed in depth elsewhere for both images[10] and 3D shapes.[3]

Normal filtering

Rather than using the bilateral filter to modify points' positions, we can instead treat Equation 1 as a spatial deformation, and update the normal n_s according to this deformation.[11] In other words, we look at how an infinitesimal patch at s with normal n_s would be modified by the filter. Since the filter removes noise from point positions, it should also align normals, as shown in Figure 1.

In a sense, the filter compresses space perpendicularly to the surface defined by the points. Points move closer to the surface, and normals align with one another. With near features, we see similar behavior, but normals align with the surface normal on one side or the other, depending on their location (see Figure 2).

Our method relies on the feature-preserving behavior of the filter with regards to geometry to preserve features in the normal field as well. This is the only requirement for feature preservation in the normals. Our normal filter inherits the feature-preserving behavior of the geometric filter it is based on.

If we write the 3D bilateral filter in Equation 1 for a point s in space as a deformation, that is, $s' = F(s)$, then the transformation of the normal can be computed from the transposed inverse of the Jacobian $J(s)$ of $F(s)$,[11]

$$n'_s = J^{-T}(s)n_s, \qquad J_i(s) = \frac{\partial F(s)}{\partial s_i}$$

where $J_i(s)$ is the ith column of $J(s)$, and s_i is the ith component of s. We have assumed that n'_s will be renormalized after smoothing, and thus use the adjoint rather than full inverse of $J(s)$. We do not give the full derivation of $J(s)$. It's straightforward to derive from Equation 1 by hand, or to use an implementation of automatic differentiation to compute it. We further explore the use of the transposed adjoint in the "Analysis of the Transposed Adjoint to the Filter" sidebar.

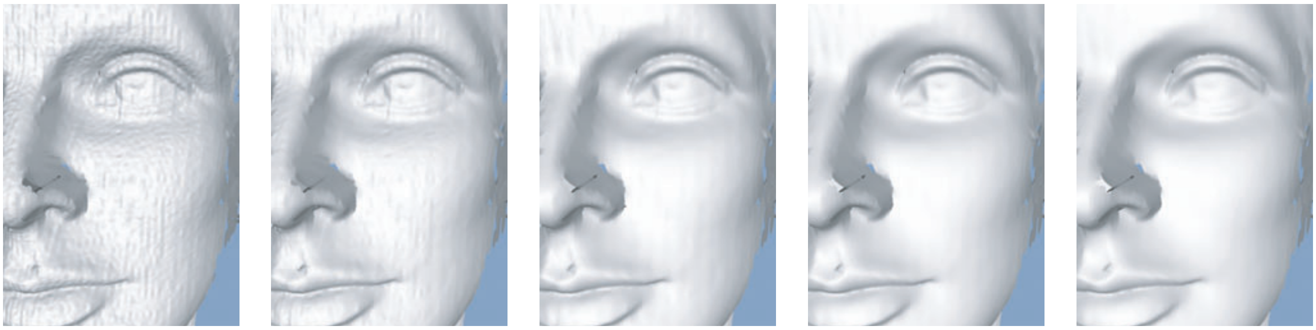
We demonstrate the results of our filter on a noisy 3D scan of a face, stored as a surfel model in Pointshop 3D (see Figure 3, next page).[12] After four iterations of the filter, nearly all noise in the normals is removed. The features, particularly around the eyes and mouth, are well preserved by the filter. The artifacts (for example, on the side of the nose) are caused by surfels with very large radii, and are present in the original data. There are about 75,000 surfels in this model. For this example, we used $\sigma_f = 4$ and $\sigma_g = 0.5$ of the mean surfel radius. In general, we must adjust the filter parameters based on sampling rate and the amount of noise in the model. The spatial filter is truncated to zero at twice σ_f. The filter runs in ≈ 30 seconds per pass on a 1.4-GHz Athlon processor. The speed could be improved through optimization. For instance, kd-trees are used to locate nearby points. Jones, Durand, and Desbrun have found it much more efficient to use spatial binning with bins of size σ_f for this purpose.[3]

We demonstrate more aggressive smoothing with $\sigma_f = 8$ on a scan of a dog (400,000 surfels.) Each pass requires ≈ 500 seconds, due to a larger model and wider filter. Figure 4 shows the initial model and two passes of the smoother.

Conclusion

We have presented a feature-preserving filter for the normals of point models. For rendering applications, this is generally sufficient, as the normals are much more important to rendering quality than the actual point positions. In simple terms, our method is a synthesis of spatial deformations[11] and feature-preserving geometry filtering.[3]

Our filter does not modify sample positions, which we believe is beneficial in an iterative smoothing process. When noisy positions and normals are filtered, some



3 **Four iterations of our filter applied to the normals of a noisy 3D scan.**

4 **Two iterations of our filter are applied to a scan of a dog statue. (Model courtesy of Jianbo Peng.)**

smoothing of features is unavoidable. By not modifying sample positions, we preserve as much information about the original features as possible, giving normals that better reflect these features.

One possibility for future work is to use our normal improvement algorithm as a presmoothing pass, then apply a filter to sample positions. We have concentrated on normals rather than positions because normals are much more important than positions in rendering quality. For other applications, such as haptics and surface reconstruction, it might be necessary to improve point positions as well.

Many scanners produce normals for samples from the positions of neighboring samples. We have not yet explored how this affects our normal smoothing method when compared to data from scanners that estimate normals from, for example, lighting variation. The models we have used to demonstrate our method were generated from triangle meshes, effectively estimating normals from neighboring positions. ■

References

1. T. Tasdizen et al., "Geometric Surface Smoothing via Anisotropic Diffusion of Normals," *Proc. IEEE Visualization*, IEEE CS Press, 2002, pp. 125-132.
2. P. Choudhury and J. Tumblin, "The Trilateral Filter for High Contrast Images and Meshes," *Proc. Eurographics Symposium on Rendering*, Eurographics Association, 2003, pp. 186-196.
3. T.R. Jones, F. Durand, and M. Desbrun, "Noniterative, Feature-Preserving Mesh Smoothing," *ACM Trans. Graphics (Proc. ACM Siggraph 03)*, vol. 22, no. 3, 2003, pp. 943-949.
4. M. Alexa et al., "Point Set Surfaces," *Proc. IEEE Visualization*, IEEE CS Press, 2001, pp. 21-28.
5. M. Pauly and M. Gross, "Spectral Processing of Point Sampled Geometry," *Proc. ACM Siggraph*, ACM Press, 2001, pp. 379-386.
6. S.M. Smith and J.M. Brady, "Susan: A New Approach to Low-Level Image Processing," *Proc. Int'l J. Computer Vision 23*, Kluwer Academic Pub., 1997, pp. 45-78.
7. C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int. Conf. Computer Vision*, IEEE CS Press, 1998, pp. 836-846.
8. S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral Mesh Denoising," *ACM Trans. Graphics (Proc. ACM Siggraph 03)*, vol. 22, no. 3, 2003, pp. 950-953.
9. H. Pfister et al., "Surfels: Surface Elements as Rendering Primitives," *Proc. ACM Siggraph*, ACM Press, 2000, pp. 335-342.
10. F. Durand and J. Dorsey, "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images," *ACM Trans. Graphics (Proc. ACM Siggraph 02)*, vol. 21, no. 3, 2002, pp. 257-266.
11. A.H. Barr, "Global and Local Deformations of Solid Primitives," *Proc. Siggraph*, ACM Press, 1984, pp. 21-30.
12. M. Zwicker et al., "Pointshop 3D: An Interactive System for Point-Based Surface Editing," *ACM Trans. Graphics (Proc. ACM Siggraph 02)*, vol. 21, no. 3, 2002, pp. 322-329.

Thouis R. Jones *is a graduate student in the MIT Computer Science and Artificial Intelligence Laboratory. His research interests include antialiasing, shape representation, and machine vision. Jones has a BS in computer science from the University of Utah.*

Frédo Durand *is an assistant professor in the Computer Science and Artificial Intelligence Laboratory at MIT. His research interests include realistic rendering, real-time display, computational photography, nonphotorealistic rendering, and perceptually based graphics. Durand has a PhD from the University of Grenoble, France, in computer science. He is a member of Eurographics and ACM Siggraph.*

Matthias Zwicker *is a postdoctoral associate with the Computer Graphics Group at MIT. His research interests include rendering algorithms and data structures for point-based surface representations, high-quality volume rendering, processing and compression of point-based data structures, acquisition of real world objects, and texturing and painting of point-sampled surfaces. Zwicker has a PhD in computer science from the Swiss Federal Institute of Technology (ETH), Zurich. He is a member Eurographics.*

Readers may contact Thouis R. Jones at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32-D474, MIT, Cambridge, MA 02139; thouis@mit.edu.