# Bilateral and Mean Shift Filtering on Irregular Domains 

Submission ID goes here


#### Abstract

Put abstract here CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling-Geometric algorithms, languages, and systems; I..4.3 [Image Processing and Computer Vision]: Enhancement-Filtering


Keywords: digital geometry processing, discrete differential geometry, geometric modeling, geometric editing, bilateral filtering, mean shift

Links: DL ${ }^{-1}$ PDF

## 1 Introduction

Signals on images, surfaces, and other domains encountered in computer graphics rarely obey the strong smoothness assumptions imposed by methods from classical signal processing. Even when these methods are successful with respect to basic measures of smoothness, continuity, or other common objectives, the resulting signal often is undesirable. For instance, Gaussian convolution has many characteristics that arguably make it the most effective image denoising filter, yet when it is applied to photographs the result disrespects object boundaries and other semantic features.

Thus, it comes as no surprise that in response to these drawbacks a plethora of nonlinear filters have been developed to take stronger priors about signal content into account. For instance, one of the simplest and most effective replacements for Gaussian convolution is the bilateral filter. Deriving the bilateral is a straightforward process: rather than blindly averaging pixels that are near each other, as in Gaussian convolution, the bilateral blends pixels that are nearby both in location and intensity. The result is a filter that behaves like Gaussian convolution within object boundaries but prevents pixels on opposite sides of a boundary from blending.

Given the success of the bilateral filter in image processing and computational photography, many attempts have been made to adapt bilateral filtering to mesh domains for edge-preserving smoothing and other tasks. Unfortunately, this transition is not a straightforward one, since the irregularity of meshed surfaces makes it difficult to evaluate the bilateral in reasonable time. Most discretizations rely on localized operations that can be sensitive to the choice of triangulation or on expensive and distortion-inducing parameterizations to reduce the mesh bilateral to the planar case. In some sense, they can be regarded as methods "inspired" by the bilateral filter rather than generalizing it that have vague if any guarantees of reasonable behavior in the limit of refinement.

In this paper, we introduce a bilateral filtering technique that can be carried out for signals on any domain admitting a heat diffusion operator. This filter coincides with the image bilateral in the planar case and when implemented with sufficient generality can be used to process signals on images, meshes, point clouds, and other domains with only a few changed lines of code. Its discretization aligns well with its continuous counterpart, and several reasonable extensions can be formulated for a larger class of filtering tasks.

One of the most important applications of the bilateral is as the main step within iterations of the mean shift filter. The mean shift is a strong denoising and edge-sharpening filter that is preferable to the bilateral in many settings. We show that the mean shift has an identical expression within our framework and even can be used to filter signals like surface normals, which most naturally should be treated as signals whose range is a subset of the unit sphere $S^{2}$ rather than $\mathbb{R}^{3}$.

Of course, on any domain, bilateral and mean shift filtering rarely are ends within themselves. Instead, they are used as components of a larger pipeline to accomplish a particular filtering task. While the theory here focuses on the development of a principled technique for generalized bilateral filtering, we show how our method applies to several tasks from geometry processing, including surface denoising, normal filtering on oriented point clouds to assist in surface reconstruction, curvature computation (HOPEFULLY), and texture smoothing (HOPEFULLY). We also (HOPEFULLY) suggest how modifications of our generalized bilateral can be used to achieve interesting non-smoothing filters that respect sharp edges.

Contributions The basic mathematical contribution of this paper is a principled framework for bilateral filtering of signals with arbitrary domain and distance manifolds, developed in Section 3. We describe a stable, easy-to-implement, and convergent discretization of this filter in Section 4, including examples of its application to signals commonly encountered in computer graphics. Section 5 develops iterative schemes for mean-shift filtering signals whose range is $\mathbb{R}^{n}$ (or a convex subset thereof) or the unit sphere $S^{2}$ using the generalized bilateral as a base, including proof that these methods are unconditionally convergent and provide strong denoising. Finally, Section 6 suggests potential extensions to our bilateral technique for applications outside of denoising.

## 2 Background

The literature on feature- and edge-preserving signal processing is vast, as is that on mesh smoothing, and we cannot summarize it here. [Sun et al. 2007; Botsch et al. 2010] contain fairly comprehensive surveys of recent work on mesh smoothing and fairing and can be used for a broader context within geometry processing, the main application we suggest for our techniques. Here, we focus on bilateral geometry filtering schemes, as they provide the closest related work to our method.

Introduced in [Tomasi and Manduchi 1998] for feature-preserving image smoothing, the bilateral filter averages signals $f: I \rightarrow \mathbb{R}^{n}$ on an image $I$ using a kernel that is the product of a spatial term $W_{s}$ and a term $W_{c}$ depending on intensity or color distance:

$$
\begin{equation*}
\bar{f}(\mathbf{x})=\frac{\int_{I} f(\mathbf{y}) W_{s}(\|\mathbf{x}-\mathbf{y}\|) W_{c}(|f(\mathbf{x})-f(\mathbf{y})|) d \mathbf{y}}{\int_{I} W_{s}(\|\mathbf{x}-\mathbf{y}\|) W_{c}(|f(\mathbf{x})-f(\mathbf{y})|) d \mathbf{y}} \tag{1}
\end{equation*}
$$

That is, pixels are combined when they are nearby both in space in intensity. Slightly generalizing this filter without affecting its computation time, the cross or joint bilateral allows filtering of one signal $f_{1}$ using intensity distances from another signal $f_{2}$ [Petschnigg et al. 2004]:

$$
\begin{equation*}
\bar{f}(\mathbf{x})=\frac{\int_{I} f_{1}(\mathbf{y}) W_{s}(\|\mathbf{x}-\mathbf{y}\|) W_{c}\left(\left|f_{2}(\mathbf{x})-f_{2}(\mathbf{y})\right|\right) d \mathbf{y}}{\int_{I} W_{s}(\|\mathbf{x}-\mathbf{y}\|) W_{c}\left(\left|f_{2}(\mathbf{x})-f_{2}(\mathbf{y})\right|\right) d \mathbf{y}} \tag{2}
\end{equation*}
$$

This way, if $f_{1}$ is too noisy or complex to have well-defined features, it still can be processed as long as $f_{2}$ is clearer. Given its pervasiveness in the image processing literature, considerable research has been put into accelerating the bilateral and cross bilateral, including [Paris and Durand 2006; Adams et al. 2009; Adams et al. 2010].
Outside of image processing, a number of methods have been developed that attempt to apply bilateral filtering to other domains. For the most part, these methods map the domain to a regular grid so that algorithms for image processing can be applied; for instance, [Miropolsky and Fischer 2004] applies the bilateral on a voxel grid for surface reconstruction. [Adams et al. 2009] can be used to process signals that are not on grids, but distances for $f_{1}$ and $f_{2}$ must be measured using the Euclidean norm $\|\cdot\|_{2}$. [Eigensatz et al. 2008] makes use of a bilateral signal for scalar curvature signals on meshes, although their main focus is on a pipeline for shape editing rather than evaluation of the bilateral itself.

One domain in which applications of the bilateral extend beyond grid-based methods is mesh fairing and smoothing. Figure 1 attempts to enumerate past approaches to extend the bilateral to mesh domains in this fashion. Despite the considerable amount of research devoted to mesh bilateral filtering (Figure 1 identifies 14 papers over less than a decade whose main contribution is a mesh bilateral smoothing filter), we find that none of the prior contributions simultaneously exhibits the following desirable properties:

1. Ability to use intrinsic and smooth distance weights such as those provided by the heat kernel without resorting to parameterization
2. An understanding of convergence in the limit of mesh refinement or a theoretical definition identifying the effects of the filter on an abstract surface
3. Reduction to the image bilateral [Tomasi and Manduchi 1998] for planar signals

## 4. Applicability to multiple signal types and domains

Our algorithm satisfies all these criteria and performs comparably to the papers in Figure 1 despite its generality.

We also show how our algorithm can be used to generalize methods for performing mean shift filters on input signals. Mean shift filtering, introduced for image segmentation in [Comaniciu and Meer 2002], was shown to be equivalent to iterated cross bilateral filtering in [Van de Weijer and Van den Boomgaard 2001] and elsewhere. Mean shift filtering is known to produce strong feature-preserving denoising in the image case, but few attempts have been made to apply it to meshes. [Yamauchi et al. 2005] mean shifts mesh normals to assist in segmentation; [Shamir et al. 2006] attempts to do so using local geodesic parameterization without acknowledging previous work on mesh bilateral filters with success filtering somewhat noisy curvatures, normals, and other shape properties.

## 3 Generalized Bilateral Filtering

Take $\Sigma$ to be the domain of a signal $f_{1}: \Sigma \rightarrow \mathbb{R}^{n}$ equipped with a symmetric kernel $K_{\Sigma}: \Sigma \times \Sigma \rightarrow \mathbb{R}$. Intuitively, we can think
of $K_{\Sigma}(\mathbf{x}, \mathbf{y})$ as measuring the proximity between $\mathbf{x}$ and $\mathbf{y}$ on $\Sigma$. For instance, signal processing on an image might take $\Sigma \subseteq \mathbb{R}^{2}$ as the image plane, $n=3$ for RGB channels, and $K_{\Sigma}(\mathbf{x}, \mathbf{y})=$ $e^{-\|\mathbf{x}-\mathbf{y}\|^{2} / \sigma^{2}}$, the usual Gaussian blur kernel. More generally, if $\Sigma$ is any domain admitting a Laplacian operator $L$, such as a graph, surface, mesh, or point cloud, we can take $K_{\Sigma}$ to be the kernel corresponding to a solution at some fixed $t>0$ of the heat equation $\frac{\partial u}{\partial t}=L u$, where $u(\mathbf{x}, t): \Sigma \times[0, \infty) \rightarrow \mathbb{R}$; that is, $K_{\Sigma}(\mathbf{x}, \mathbf{y})$ measures how much a unit of heat diffuses from $\mathbf{x}$ to $\mathbf{y}$ along $\Sigma$ in $t$ time.
With the kernel $K_{\Sigma}$, we can define a blurred version of $f_{1}$ as the convolution

$$
\begin{equation*}
\hat{f}_{1}(\mathbf{x})=\frac{1}{z(\mathbf{x})} \int_{\Sigma} f_{1}(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) d \mathbf{y} \tag{3}
\end{equation*}
$$

where $z(\mathbf{x})$ is the normalizing value $\int_{\Sigma} K_{\Sigma}(\mathbf{x}, \mathbf{y}) d \mathbf{y}$; usually $z$ is constant, but for our construction this restriction is unnecessary. Define the functional $T_{K}: L^{2}(\Sigma) \rightarrow L^{2}(\Sigma)$ such that $T_{K}(f)=\hat{f}$ defined above; note that $T_{K}$ simply is the linear operator blurring out $f$ with kernel $K$.

Now, in parallel with the development with the image cross bilateral filter (2), take $f_{2}: \Sigma \rightarrow \Gamma$ to be a cross bilateral function designed so that if $f_{2}(\mathbf{x})$ and $f_{2}(\mathbf{y})$ are very different, the signal $f_{1}$ at $\mathbf{x}$ and y should not be blended during filtering. We assume that $\Gamma$ is a compact manifold with boundary; for instance, using RGB colors in $[0,1]$ for the cross bilateral function would yield $\Gamma=[0,1]^{3}$, while using surface normals yields $\Gamma=S^{2} \subset \mathbb{R}^{3}$, the unit sphere. We equip $\Gamma$ with its own kernel $K_{\Gamma}$.
With this notation in place, we can introduce the generalized crossbilateral filter as follows:

$$
\begin{equation*}
\bar{f}(\mathbf{x})=\frac{\int_{\Sigma} f_{1}(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f_{2}(\mathbf{x}), f_{2}(\mathbf{y})\right) d \mathbf{y}}{\int_{\Sigma} K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f_{2}(\mathbf{x}), f_{2}(\mathbf{y})\right) d \mathbf{y}} \tag{4}
\end{equation*}
$$

Note the similarity to the image cross bilateral filter (2). The main difference is that we allow our kernel functions to take into account $\mathbf{x}$ and $\mathbf{y}$ (as well as $f_{2}(\mathbf{x})$ and $f_{2}(\mathbf{y})$ ) directly rather than just the norms $\|\mathbf{x}-\mathbf{y}\|$ (and $\left.\left\|f_{2}(\mathbf{x})-f_{2}(\mathbf{y})\right\|\right)$, which may not be welldefined depending on the choice of $\Sigma$ and $\Gamma$.
Note that we can re-express the cross bilateral using the diffusion operator $T_{K}$ defined above. In particular, define numerator and denominator functions as:

$$
\begin{align*}
f_{\mathbf{p}}^{n u m}(\mathbf{y}) & =f_{1}(\mathbf{y}) K_{\Gamma}\left(f_{2}(\mathbf{y}), \mathbf{p}\right)  \tag{5}\\
f_{\mathbf{p}}^{\text {den }}(\mathbf{y}) & =K_{\Gamma}\left(f_{2}(\mathbf{y}), \mathbf{p}\right) \tag{6}
\end{align*}
$$

Then, we have

$$
\begin{equation*}
\bar{f}(\mathbf{x})=\frac{T_{K}\left[f_{f_{2}(\mathbf{x})}^{n u m}(\cdot)\right](\mathbf{x})}{T_{K}\left[f_{f_{2}(\mathbf{x})}^{d e n}(\cdot)\right](\mathbf{x})} \tag{7}
\end{equation*}
$$

## 4 Discretization

We use a signal processing technique similar to that in [Paris and Durand 2006] to evaluate the bilateral filter on discrete domains using the expression (7). Since the same computations apply to $f_{\mathbf{p}}^{\text {den }}$ as $f_{\mathbf{p}}^{\text {num }}$ after replacing $f_{1}$ with 1 , for ease of notation denote $f_{\mathrm{p}}$ as one of $f_{\mathrm{p}}^{\text {num }}$ or $f_{\mathbf{p}}^{d e n}$; our algorithm evaluates each separately and performs the division as a final pass.
Suppose that we choose samples $\mathbf{p}_{1}, \ldots, \mathbf{p}_{m} \in \Gamma$ and a corresponding partition of unity $\phi_{1}, \ldots, \phi_{m} \in L^{2}(\Gamma)$ such a function

| Paper | Description |
| :--- | :--- |
| [Fleishman et al. 2003] | Bilateral filters the height function of the surface over vertex tangent planes |
| [Jones et al. 2003] | Combines vertices with their projections onto nearby tangent planes with bilateral weights from dis- <br> tance to the tangent plane projection and the tangent plane center |
| [Hu et al. 2004] | Uses bilateral filtering as part of a multi-pass approach to modify Laplacian smoothing using weights <br> inspired by those in [Fleishman et al. 2003] |
| [Jones et al. 2004] | Iteratively applies a modification of [Jones et al. 2003] to improve surface normals for rendering. |
| [Duguet et al. 2004] | Bilateral filters jets on point clouds for reconstruction |
| [Hou et al. 2005] | Bilateral filters mesh normals and then adjusts surface; weights are Gaussians in normal difference |
| and an approximation of geodesic distance |  |
| [Shimizu et al. 2005] | Explicitly filters sharp edges and then faces separately using extrinsic distances, edge directions, <br> normal difference, and projections as in [Jones et al. 2003] <br> [Lee and Wang 2005] <br> [Wang 2006] <br> [Adams et al. 2009] <br> Filters face normals using Euclidean distance between centroids and normal differences <br> [Fan et al. 2010] <br> [Nociar and Ferko 2010] <br> Filters non-manifold surfaces by using iteratively applying a bilateral similar to [Jones et al. 2003] <br> [Zheng et al. 2011] <br> and remeshing |
| [Vialaneix and Boubekeur 2011] | Filters the difference between a mesh and its Laplace-smoothed counterpart in principal curvature <br> coordinates using spin-images [Johnson and Hebert 1999] for weights without a distance term |

Figure 1: A summary of previous attempts to adapt bilateral filtering to mesh domains.
$g: \Gamma \rightarrow \mathbb{R}$ can be approximated as $g(\mathbf{p}) \approx \sum_{i} g\left(\mathbf{p}_{i}\right) \phi_{i}(\mathbf{p})$. Note that under mild continuity and compact support conditions, we can construct sequences of partitions such that the approximation converges to $g(\mathbf{p})$ as $m \rightarrow \infty$.

```
Input : Signal to be filtered \(f_{1}: \Sigma \rightarrow \mathbb{R}^{n}\)
    Distance function \(f_{2}: \Sigma \rightarrow \Gamma\)
    Samples \(\mathbf{p}_{1}, \ldots, \mathbf{p}_{m} \in \Gamma\)
    Partition of unity \(\phi_{1}, \ldots, \phi_{m} \in L^{2}(\Gamma)\)
Output: Filtered signal \(\bar{f}: \Sigma \rightarrow \mathbb{R}^{n}\)
\(\bar{f}^{\text {num }}(\mathbf{x}), \bar{f}^{\text {den }}(\mathbf{x}) \leftarrow 0 \forall \mathbf{x} \in \Sigma \quad\) Initialization
for \(i=1\) to \(m \mathbf{d o}\)
    \(\hat{g}^{\text {num }}(\mathbf{x}) \leftarrow f_{1}(\mathbf{x}) K_{\Gamma}\left(f_{2}(\mathbf{x}), \mathbf{p}_{i}\right) \quad\) Weight signals
    \(\hat{g}^{\text {den }}(\mathbf{x}) \leftarrow K_{\Gamma}\left(f_{2}(\mathbf{x}), \mathbf{p}_{i}\right)\)
    \(g^{\text {num }}(\mathbf{x}) \leftarrow T_{K}\left[\hat{g}^{\text {num }}\right](\mathbf{x}) \quad\) Apply blur operator
    \(g^{d e n}(\mathbf{x}) \leftarrow T_{K}\left[\hat{g}^{d e n}\right](\mathbf{x})\)
    \(\bar{f}^{\text {num }}(\mathbf{x}) \leftarrow g^{\text {num }}(\mathbf{x}) \phi_{i}\left(f_{2}(\mathbf{x})\right) \quad\) Collect
    \(\bar{f}^{d e n}(\mathbf{x}) \leftarrow g^{d e n}(\mathbf{x}) \phi_{i}\left(f_{2}(\mathbf{x})\right)\)
end
\(\bar{f}(\mathbf{x}) \leftarrow \bar{f}^{n u m}(\mathbf{x}) / \bar{f}^{\text {den }}(\mathbf{x}) \quad\) Normalize
```

Algorithm 1: Generalized bilateral filtering algorithm

With this partition in place, our algorithm becomes fairly straightforward to describe. Define $\hat{g}_{i}(\mathbf{x})=f_{\mathbf{p}_{i}}(\mathbf{x})$; note that this function can be computed $\forall \mathbf{x} \in \Sigma$ simply by evaluating $f_{1}$ and $K_{\Gamma}$ as in (5) and (6). Then, apply the blurring operation (3) to obtain $g_{i}(\mathbf{x})=T_{K}\left[\hat{g}_{i}\right](\mathbf{x})$. Our bilateral filter is thus approximated as:

$$
\begin{equation*}
\bar{f}(\mathbf{x}) \approx \frac{\sum g_{i}^{\text {num }}(\mathbf{x}) \phi_{i}\left(f_{2}(\mathbf{x})\right)}{\sum g_{i}^{\text {den }}(\mathbf{x}) \phi_{i}\left(f_{2}(\mathbf{x})\right)} \tag{8}
\end{equation*}
$$

Our algorithm is summarized in Algorithm 1.
With the generalized filtering method in place, we proceed to show several concrete applications of bilateral filtering simply by defining the relevant domains, kernels, and operators. Notice that if $K_{\Gamma}$ is straightforward to evaluate, the only time-consuming step is gen-
erating the functions $g_{i}$ from $\hat{g}_{i}$; that is, the time complexity of this algorithm is essentially that of carrying out $m$ blurs (3).

### 4.1 Grayscale Image Signals

Before proceeding to novel domains and signals, we verify that our algorithm applied to grayscale images reduces to the one presented in [Paris and Durand 2006] without down- and up-sampling. Here, we define our signal domain as $\Sigma=\{1, \ldots, w\} \times\{1, \ldots, h\}$, a $w \times h$ grid of pixel values, and our signal range of grayscale intensities is $\Gamma=[0,1]$. We take our image and intensity kernels to be $K_{\Sigma}(\mathbf{x}, \mathbf{y}) \equiv W_{s}(\|\mathbf{x}-\mathbf{y}\|)$ and $K_{\Gamma}(p, q)=W_{c}(\mid p-q \|)$. It is easy to check that in this case (4) and (2) coincide.

Now, suppose we divide $\Gamma=[0,1]$ into $m$ equally-spaced samples $p_{1}, \ldots, p_{m}$ of width $1 / m-1$. Define $\phi_{i}:[0,1] \rightarrow \mathbb{R}$ to be the piecewise linear hat function centered at $p_{i}$ with width $2 / m-1$. Then, (8) coincides with the "signal processing approximation" in [Paris and Durand 2006]. The approximation is virtually indistinguishable from the exact bilateral filter on most images for $m$ as low as 20 , and it can be carried out efficiently using downsampling and fast Gaussian convolution such as [Burt and Adelson 1983; Deriche 1993] for (3).

### 4.2 Scalar Mesh Signals

Now, suppose we take $\Sigma$ to be a mesh $(V, E, F)$ with vertices $V$, edges $E$, and triangular faces $F$. We represent functions on $\Sigma$ as vectors $\mathbf{v} \in \mathbb{R}^{|V|}$, with one value per vertex, and can construct a "cotangent Laplacian" matrix $L \in \mathbb{R}^{|V| \times|V|}$ imitating the Laplacian operator on the smooth surface approximated by $\Sigma$ [Botsch et al. 2010]. We use compute $T_{K}(\mathbf{v})$ as heat flow using a single implicit time step $T_{K}(\mathbf{v}) \approx(I+\Delta t \cdot L)^{-1} \mathbf{v}$; multiple time steps could yield closer approximations, but the damping effect of a single implicit step has few perceptual differences and is faster to carry out. Since we will have to apply $T_{K}$ several times, we pre-factor $I+\Delta t \cdot L$ using the sparse LU method in [Davis 2004]. We keep $\Gamma=[0,1]$ with Gaussian kernel $K_{\Gamma}(p, q)=e^{-|p-q|^{2} / \sigma^{2}}$.

Now, if we take $f_{1}=f_{2} \equiv \mathbf{v}$, the bilateral filter in Algorithm 1 blurs out signals on $\Sigma$ while preserving intensity edges in $\mathbf{v}$. Figure NUMBER shows the output of such a process on assorted signals on surfaces. Note that unlike the image bilateral or mesh methods relying on planar projection or parameterization, this bilateral respects the metric of $\Sigma$ regardless of the width of $K_{\Sigma}$.
[discuss curvature filtering here if it works]

### 4.3 Normal Vector Mesh Signals

We can extend the method in Section 4.2 above by considering cross bilaterals for which the domain $\Gamma$ of $f_{2}$ is not $[0,1]$. Perhaps most importantly, suppose $\Gamma=S^{2}$, the unit sphere, and take $f_{2}$ to be signal $\mathbf{N}: F \rightarrow S^{2}$ of unit face normals. Note that our signal now is on mesh faces rather than vertices to avoid ambiguous normals along sharp edges, so we replace $L$ from 4.2 with the dual 0 -form Laplacian $d \star d \star$ defined within the framework of discrete exterior calculus [Hirani 2003].

We equip $\Gamma=S^{2}$ with the Von Mises-Fisher kernel $K_{\Gamma}(\mathbf{p}, \mathbf{q})=$ $e^{\mathbf{p} \cdot \mathbf{q} / \sigma}$, used to represent isotropic distributions on the unit sphere [Fisher 1953]. A convergent partition of unity on $S^{2}$ is obtained using a regular unit-radius polyhedron inscribed within $S^{2}$; each $\phi_{i}$ corresponds to a picewise linear hat function centered at a vertex of the polyhedron, projected to the surface of the sphere. An alternative more efficient and smoother partition of unity with less clear convergence guarantees is to simply use Von Mises-Fisher kernels centered at sample points distributed around the unit sphere normalized at each point on $S^{2}$ to sum to 1 ; we find little qualitative difference between these two approaches in practice. Applications of this filter to scalar functions on $\Sigma$ are shown in Figure NUMBER. Note that the scalar values are not combined over sharp edges since the normal function $\mathbf{N}$ has a discontinuity there.

If we filter the $x, y$, and $z$ components of the signal $\mathbf{N}$ itself and renormalize, we obtain a denoised normal vector signal over $\Sigma$. This method is a direct analog of the normal smoothing methods in [Zheng et al. 2011], which can be considered approximations of the filter for small blending radii. Thus, as in [Zheng et al. 2011], we can adjust the surface of $\Sigma$ using the iterative scheme in [Sun et al. 2007] to match the denoised normals. Figure NUMBER compares denoising results of the filter at hand with the output of comparable methods.

## [discuss mean curvature normal filtering here if it works]

### 4.4 Oriented Point Cloud

Algorithms such as [Kazhdan et al. 2006] for surface reconstruction rely on oriented point clouds, which contain both sample points and their normals, to generate meshed surfaces; the normals implicitly help decipher tangent directions, orientation, and connectivity from the cloud. Methods for obtaining or computing orientations often yield noisy normals at best, which, combined with already noisy point clouds, can lead to topological and geometric reconstruction errors that can be difficult to correct during post-processing.

Fortunately, [Belkin et al. 2009] introduces a Laplacian operator for signals on point clouds with provable convergence. Combining Laplacian heat diffusion with the bilateral term ensures that edges are preserved and that the topology of the surface is respected while combining "nearby" normals. Figure NUMBER shows examples of reconstruction using [Kazhdan et al. 2006] with and without bilateral normal filtering.

### 4.5 Other Signals

Although we choose to focus on them here, the bilateral filters we discuss above are by no means the only ones that could be considered in our framework. Additional domains to which we could apply the algorithm in Algorithm 1 include:

- Mesh textures equipped with a blurring operator either from texture pyramids such as MIP maps or, to better respect the surface metric, using a Laplacian $\Delta$ pulled back from a triangulated surface
- Graphs with discrete Laplacian matrices
- Range images with colors or normals as cross bilateral signals
- Volumetric signals with heat flow using the signal as a density
- Simplicial complexes with combinatorial Laplacian heat flow

Many of the above applications are outside the realm of computer graphics; others may not benefit as much from a bilateral filter as from related techniques suggested by our method, like that for computing local histograms in Section NUMBER.

## 5 Mean Shift Filtering

### 5.1 Signals on $\mathbb{R}^{n}$

Bilateral filtering is a reliable tool for minor denoising but is less effective for highly-noisy signals. In particular, the $K_{\Gamma}$ term is designed to combine values only when they are similar; outliers thus will be influenced only slightly by their nearby counterparts. Furthermore, in certain denoising scenarios it is desired not only to smooth out signals but also to sharpen edges, either for simplification purposes or to remove softness that might have been added by certain noise models.

We can re-examine the definition of the bilateral to achieve stronger filtering. In particular, for fixed $\mathrm{x} \in \Sigma$, we refashion the denominator of the bilateral as a probability distribution over $\Gamma$ (define $f \equiv f_{2}$ ):

$$
\begin{equation*}
h(\mathbf{p} ; \mathbf{x})=\frac{1}{z(\mathbf{x})} \int_{\Sigma} K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}(\mathbf{p}, f(\mathbf{y})) d \mathbf{y} \tag{9}
\end{equation*}
$$

where $z(\mathbf{x})$ is a constant value guaranteeing $\int_{\Gamma} h(\mathbf{p} ; \mathbf{x}) d \mathbf{p}=1$. This function, constructed using the Parzen window technique as in [Kass and Solomon 2010], represents the distribution of values of $f$ near $\mathbf{x}$.
If we take $\Gamma=\mathbb{R}^{n}$ equipped with $K_{\Gamma}(\mathbf{p}, \mathbf{q})=e^{-\|\mathbf{p}-\mathbf{q}\|^{2} / \sigma^{2}}$ and take the gradient with respect to $\mathbf{p}$, we find that peaks $\mathbf{p}^{*}$ of $h(\mathbf{p} ; \mathbf{x})$ satisfy

$$
\begin{equation*}
\mathbf{p}^{*}=\frac{\int_{\Sigma} f(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(\mathbf{p}^{*}, f(\mathbf{y})\right) d \mathbf{y}}{\int_{\Sigma} K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(\mathbf{p}^{*}, f(\mathbf{y})\right) d \mathbf{y}} \tag{10}
\end{equation*}
$$

This relationship suggests a fixed-point iteration scheme for finding peaks of $h(\mathbf{p} ; \mathbf{x})$ at all $\mathbf{x}$ :

$$
\begin{align*}
f^{(0)}(\mathbf{x}) & =f(\mathbf{x})  \tag{11}\\
f^{(k+1)}(\mathbf{x}) & =\frac{\int_{\Sigma} f(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f^{(k)}(\mathbf{x}), f(\mathbf{y})\right) d \mathbf{y}}{\int_{\Sigma} K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f^{(k)}, f(\mathbf{y})\right) d \mathbf{y}} \tag{12}
\end{align*}
$$

Each iteration simply is an application of the cross bilateral filter (4). This iterative scheme is known as the mean-shift filter [Comaniciu and Meer 2002], and with some modification can be ap-
plied to other choices of $K_{\Gamma}$. It is known to converge unconditionally to peaks of $h$, as proved in [Li et al. 2007], and is a wellunderstood object in both image processing and machine learning. Figure NUMBER shows examples of the mean shift on scalar signals on triangle meshes; note the strong denoising that occurs relative to the bilateral.

### 5.2 Signals on $S^{2}$

The application of the bilateral to filter mesh and point cloud normals in Sections 4.3 and 4.4 has a serious drawback: before normalization, the output of the bilateral is unlikely to have unit length. That is, bilateral filtering inputs signals on $S^{2}$ but outputs signals in $\mathbb{R}^{3}$. This property is a drawback of several surface bilateral filtering methods, including CITE, making the behavior of these filters difficult to understand and control.

The construction of $h$ in Section 5.1, however, remains valid when $\Gamma$ is not $R^{n}$ (or a convex subset thereof). Using this observation as a starting point, in Appendix NUMBER we show that paralleling the derivation of the Euclidean mean shift algorithm using the Von Mises-Fisher distribution on $S^{2}$ yields the iterative scheme

$$
\begin{align*}
f^{(0)}(\mathbf{x}) & =f(\mathbf{x})  \tag{13}\\
f^{(k+1)}(\mathbf{x}) & =\frac{\int_{\Sigma} f(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f^{(k)}(\mathbf{x}), f(\mathbf{y})\right) d \mathbf{y}}{\left\|\int_{\Sigma} f(\mathbf{y}) K_{\Sigma}(\mathbf{x}, \mathbf{y}) K_{\Gamma}\left(f^{(k)}(\mathbf{x}), f(\mathbf{y})\right) d \mathbf{y}\right\|} \tag{14}
\end{align*}
$$

Note that each iterate remains on $S^{2}$, thus allowing the filter to be regarded as one for signals on $S^{2}$ rather than its embedding in $\mathbb{R}^{3}$. The proof of Euclidean mean-shift convergence from [Li et al. 2007] no longer holds with this modification, however, thanks to the renormalization that occurs every iteration. Fortunately, this new iterative scheme is an instance of the spherical mean shift algorithm in [Kobayashi and Otsu 2010], which proves its convergence and qualitatively similar behavior to the Euclidean case.

While a single iteration of the bilateral filter has moderate denoising ability, the mean shift on $S^{2}$ is a powerful edge-preserving denoising filter. Figures NUMBER and NUMBER show denoising results on meshes and oriented point clouds.

## 6 Extensions

Up to this point, the primary application we have suggested of our filtering technique is smoothing of signals on arbitrary domains with arbitrary similarity measures. Here, we show that as with the image bilateral, our method can be applied to other problems in mesh processing and other domains.

### 6.1 Local Histograms

[Kass and Solomon 2010] suggests that the histogram $h(\mathbf{p} ; \mathbf{x})$ defined in (9) has value by itself for understanding a given signal; in particular, they use this function to understand the distribution of pixel intensities in some smoothly-weighted neighborhood of each pixel, allowing for direct rather than iterative evaluation of the median, mean shift, global mode, local histogram equalization, and other filters. An identical formulation applies to our more general setting. In particular, the task of evaluating $h\left(\mathbf{p}_{i} ; \mathbf{x}\right) \forall \mathbf{x} \in \Sigma$ is identical to the evaluation of the samples in the denominator in Algorithm 1. Thus, we can efficiently extract local histograms of signals on $\Sigma$ with value in compact manifold $\Sigma$. This extension allows for the direct evaluation of the same filters applied to scalar functions on surfaces and other domains.

The method at our level of generality, however, can be applied to a much wider array of signals. In particular, once again taking $f: \Sigma \rightarrow \Gamma=S^{2}$ to be the normal vector signal on $\Sigma$, the histogram $h(\mathbf{p} ; \mathbf{x})$ at a fixed $\mathbf{x} \in \Sigma$ now represents the distribution over $S^{2}$ of normal vectors to $\Sigma$ near $\mathbf{x}$. This distribution can be viewed (after suitable rotation) as a version of the SHOT descriptor introduced in [Tombari et al. 2010] with smoothly varying, intrinsic heat kernel weights on $\Sigma$ rather than extrinsic distance weights, with straightforward regularization control by changing blurring radii on $\Sigma$ and $\Gamma$. Figure NUMBER shows some examples of normal vector histograms computed using this technique.

### 6.2 Feature-Preserving Filters

We have gone a long way toward pushing the bilateral filter to a maximal of generality. One additional avenue for flexibility, however, is in the choice of kernels $K_{\Sigma}$ and $K_{\Gamma}$.

The most obvious potential change in $K_{\Sigma}$ or $K_{\Gamma}$ might be in the choice of smoothing kernels. We implicitly have made use of this possibility by suggesting that a single implicit time step of the heat equation suffices for bilateral filtering on meshes. In practice, we find that any reasonable choice of smoothing kernel behaves in a qualitatively similar fashion for most bilateral and mean shift applications.

Even more generally, heat flow is one of a huge class of linear operators used in mesh processing. Band-pass, high-pass, unsharp mask, and other filters can be applied to signals on a surface using analogs of Fourier theory and a discretization of the Laplacian. Even if these filters are described using some sort of local operation, their linearity implies the existence of an operator matrix, which in turn contains kernel values $K_{\Sigma}: \mathbb{R}^{|V|} \times \mathbb{R}^{|V|} \rightarrow \mathbb{R}$. In the continuous limit, the theory of Schwartz kernels guarantees that most wellposed linear operators on surface signals admit kernels, although they must be regarded as distributions to be integrated against rather than functions in the elementary sense. Our theoretical bilateral filter (4) continues to make sense in this measure-theoretic context, however, and can be regarded as a way to reweight the kernel to respect edges.

Although fully exploring the domain of feature-preserving mesh editing operations is worthy of a larger study, Figure NUMBER shows some examples of the application of our bilateral (4) where the spatial kernel $K_{\Sigma}$ has been replaced with the kernels of other linear operators. We (HOPEFULLY) can achieve a number of interesting effects that respect mesh edges with no more computational complexity than our original bilateral.

## 7 Discussion

We have written a simple implementation of our algorithm in $\mathrm{C}++$, taking full advantage of templates to encode Algorithm 1 in full generality; we use some simple OpenMP directives to achieve parallel evaluation of the blurs needed for each sample $\mathbf{p}_{i}$. On a MACHINE INFO with NUMBER cores, this naïve implementation is able to apply bilateral filters to mesh normals on NUMBER faces in TIME seconds using NUMBER sample points on $S^{2}$. Subsequent iterations for the mean shift are even faster, since the heat flow matrix needs to be factored only once; NUMBER iterations on the same mesh takes TIME seconds in total.

We believe even faster runtimes could be achieved with an optimized implementation and careful evaluation of potential linear solvers. Theoretically, our runtime is approximately equal to the time it takes to blur $m$ signals using $K_{\Sigma}$, so fewer samples $\mathbf{p}_{i}$ on $S^{2}$ directly make for better timings; we can cut our number of sam-
ples to half of the ones listed here with reasonable effect but slight visible artifact in exchange for a faster filter.
(HOPEFULLY) Although it is impossible to verify our algorithm against all previous work, Figure NUMBER attempts to compare against some recent approaches. We apply Gaussian noise of varying sizes to different meshes and then apply our and other methods for smoothing to attempt to recover the original shape. We show the perceptual distance between the filtered signals and the original Jusing the perceptual metric in [Váša and Petrík 2011]. In general, we find SUMMARIZE RESULTS.

## 8 Conclusion

If nothing else, the sheer number of attempts to discretize bilateral filtering on non-image domains illustrated in Figure 1 demonstrates the elusiveness and importance of a generalized bilateral filter. Expressions for the bilateral, whether for images as in (2) or in the more general sense as in (4), are easy to state and understand and have only a few intuitive parameters. The bilateral's behavior is well-understood and forms the basis for more complex methods such as the mean shift. It has withstood the test of time and remains a foundational tool used to construct state-of-the-art algorithms in diverse parts of image processing, vision, and graphics.

Our new discretization makes the process of defining a bilateral filter on a given domain and signal straightforward. Featurepreserving filters can be achieved on arbitrary domains simply by choosing domains $\Sigma, \Gamma$ and reasonable kernels $K_{\Sigma}, K_{\Gamma}$, with the assumption that $\Gamma$ can be sampled reasonably. This process has an easily-understood continuous limit (4) and can even be extended to tasks like histogram computation and shape editing. The speed of the filter simply depends on the number of samples in $\Gamma$ and the time it takes to apply $K_{\Sigma}$, the latter of which often boiling down to a simple pre-factored linear solve.

While we have illustrated only a few applications of our method within the domain of geometry processing, we hope that its simplicity and effectiveness will lead it to be applied to other problems. For instance, in image processing, some results show that distances between some signatures we use for some commonly-used cross bilateral signals should not be measured using the Euclidean metric but rather along some underlying manifold [Carlsson et al. 2008]; this type of relationship can be encoded in our framework by defining $\Sigma$ to be a part of the image plane and $\Gamma$ to be the cross bilateral manifold in question. As another example, local histograms may be useful for understanding structure and local information in graphs, using Laplacian heat flow to evaluate proximity. These broad applications and many others are no harder to implement or understand than the ones we have suggested in this paper and begin to reveal the exciting potential implications of a reliable generalized bilateral filtering technique.

## Acknowledgments [Removed for paper submission]

## References

Adams, A., Gelfand, N., Dolson, J., and Levoy, M. 2009. Gaussian KD-trees for fast high-dimensional filtering. ACM Trans. Graph. 28, 21:1-21:12.

Adams, A., Baek, J., and Davis, M. A. 2010. Fast highdimensional filtering using the permutohedral lattice. In Comp. Graph. Forum, vol. 29, 753-762.

Belkin, M., Sun, J., and Wang, Y. 2009. Constructing Laplace Operator from Point Clouds in $\mathbb{R}^{d}$. In Proc. Symp. Disc. Alg., 1031-1040.

Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Lévy, B. 2010. Polygon Mesh Processing. AK Peters.

Burt, P., and Adelson, E. 1983. The Laplacian Pyramid as a Compact Image Code. IEEE Trans. Comm. 31, 4, 532-540.

Carlsson, G., Ishkhanov, T., Silva, V., and Zomorodian, A. 2008. On the Local Behavior of Spaces of Natural Images. Int. J. Comput. Vision 76 (January), 1-12.

Comaniciu, D., and Meer, P. 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. \& Mach. Intel. 24, 5, 603-619.
Davis, T. 2004. Algorithm 832: UMFPACK V4.3-an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. 30 (June), 196-199.
Deriche, R. 1993. Recursively Implementing the Gaussian and its Derivatives. Tech. Rep. 1893, INRIA.
Duguet, F., Durand, F., and Dettrakis, G., 2004. Robust Higher-Order Filtering of Points.

Eigensatz, M., Sumner, R. W., and Pauly, M. 2008. Curvature-Domain Shape Processing. Comp. Graph. Forum 27, 2, 241-250.

Fan, H., Yu, Y., and Peng, Q. 2010. Robust Feature-Preserving Mesh Denoising Based on Consistent Subneighborhoods. IEEE Trans. Vis. Comp. Graph. 16, 2, 312-324.

Fisher, R. 1953. Dispersion on a Sphere. Proc. Royal Society of London 217, 1130, 295-305.
Fleishman, S., Drori, I., and Cohen-Or, D. 2003. Bilateral mesh denoising. ACM Trans. Graph. 22, 950-953.

Hirani, A. 2003. Discrete Exterior Calculus. PhD thesis, Caltech.
Hou, Q., Bai, L., and Wang, Y. 2005. Mesh Smoothing via Adaptive Bilateral Filtering. In ICCS, vol. 3515, 273-280.
Hu, G.-F., Peng, Q.-S., and Forrest, A. R. 2004. Robust mesh smoothing. J. Comp. Sci. Technol. 19, 521-528.

Johnson, A. E., and Hebert, M. 1999. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. IEEE Trans. Pattern Anal. \& Mach. Intel. 21, 433-449.

Jones, T. R., Durand, F., and Desbrun, M. 2003. Noniterative, feature-preserving mesh smoothing. ACM Trans. Graph. 22, 943-949.

Jones, T., Durand, F., and Zwicker, M. 2004. Normal improvement for point rendering. Comp. Graph. and App. 24, 4, 53-56.

Kass, M., and Solomon, J. 2010. Smoothed Local Histogram Filters. ACM Trans. Graph. 29, 4, 100:1-100:10.

Kazhdan, M., Bolitho, M., and Hoppe, H. 2006. Poisson Surface Reconstruction. In Proc. Symp. Geom. Proc., 61-70.

Kobayashi, T., and Otsu, N. 2010. Von Mises-Fisher Mean Shift for Clustering on a Hypersphere. In Int. Conf. Pattern Rec., 2130-2133.

Lee, K.-W., and Wang, W.-P. 2005. Feature-Preserving Mesh Denoising via Bilateral Normal Filtering. In Proc. Comp. Aided Design and Comp. Graph, 275-280.

Li, X., Hu, Z., and Wu, F. 2007. A note on the convergence of the mean shift. Pattern Recognition 40, 6, 1756-1762.

Miropolsky, A., and Fischer, A. 2004. Reconstruction with 3D geometric bilateral filter. In Symp. Solid Modeling and App., Eurographics Association, 225-229.

Nociar, M., and Ferko, A. 2010. Feature-preserving mesh denoising via attenuated bilateral normal filtering and quadrics. In Proc. Spring Conf. on Comp. Graph., 149-156.

Paris, S., and Durand, F. 2006. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. In Proc. European Conf. Comp. Vis., 568-580.

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K. 2004. Digital photography with flash and no-flash image pairs. ACM Trans. Graph. 23, 664-672.

Shamir, A., Shapira, L., and Cohen-Or, D. 2006. Mesh analysis using geodesic mean-shift. Vis. Comp. 22, 99-108.
Shimizu, T., Date, H., Kanai, S., and Kishinami, T. 2005. A New Bilateral Mesh Smoothing Method by Recognizing Features. In Proc. Comp. Aided Design and Comp. Graph., 281286.

Sun, X., Rosin, P., Martin, R., and Langbein, F. 2007. Fast and Effective Feature-Preserving Mesh Denoising. IEEE Trans. Vis. Comp. Graph. 13, 5, 925 -938.
Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. 839-846.
Tombari, F., Salti, S., and Di Stefano, L. 2010. Unique signatures of histograms for local surface description. In Proc. European Conf. Comp. Vis., 356-369.

Van de Weijer, J., and Van den Boomgatrd, R. 2001. Local mode filtering. vol. 2, 428-433.
VÁŠA, L., AND PeTŘík, O. 2011. Optimising Perceived Distortion in Lossy Encoding of Dynamic Meshes. Comp. Graph. Forum 30, 5, 1439-1449.

Vialaneix, G., and Boubekeur, T. 2011. SBL Mesh Filter: A Fast Separable Approximation of Bilateral Mesh Filtering. In Vision, Modeling and Visualization.

WANG, C. 2006. Bilateral recovering of sharp edges on featureinsensitive sampled meshes. IEEE Trans. Vis. Comp. Graph. 12, 4, 629-639.

Yamauchi, H., Lee, S., Lee, Y., Ohtake, Y., Belyaev, A., and Seidel, H.-P. 2005. Feature Sensitive Mesh Segmentation with Mean Shift. In Proc. Shape Modeling and App., 238-245.
Zheng, Y., Fu, H., Au, O. K.-C., and Tai, C.-L. 2011. Bilateral Normal Filtering for Mesh Denoising. IEEE Trans. Vis. Comp. Graph. 17, 10, 1521-1530.

