A General Framework for Bilateral and Mean Shift Filtering

Justin Solomon, Keenan Crane, Adrian Butscher, and Chris Wojtan

Abstract—We present a generalization of the bilateral filter that can be applied to feature-preserving smoothing of signals on images, meshes, and other domains within a single unified framework. Our discretization is competitive with state-of-the-art smoothing techniques in terms of both accuracy and speed, is easy to implement, and has parameters that are straightforward to understand. Unlike previous bilateral filters developed for meshes and other irregular domains, our construction reduces *exactly* to the image bilateral on rectangular domains and comes with a rigorous foundation in both the smooth and discrete settings. These guarantees allow us to construct unconditionally convergent mean-shift schemes that handle a variety of extremely noisy signals. We also apply our framework to geometric edge-preserving effects like feature enhancement and show how it is related to local histogram techniques.

◆

1 INTRODUCTION

Signals on images, surfaces, and other domains rarely obey the smoothness assumptions imposed by methods from classical signal processing. Even when these methods are successful with respect to formal measures like smoothness and continuity, the resulting signal may fail to meet basic aesthetic or perceptual criteria. For instance, Gaussian convolution is arguably an ideal image denoising filter, yet it ignores object boundaries and other semantic features.

As a result, a variety of nonlinear filters have been developed to take priors on signal content into account. In particular, an effective replacement for Gaussian convolution is the *bilateral filter*: rather than blindly averaging pixels that are near each other, the bilateral blends pixels that are nearby in both 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 averaging together.

Due to the success of the bilateral in image processing and computational photography, many attempts have been made to adapt it to geometric domains like meshes. This transition is not straightforward, however: existing discretizations rely on local operations that are sensitive to the triangulation or use a distortion-inducing parameterization. In some sense these methods are only "inspired" by the bilateral filter and provide few guarantees in the limit of refinement.

We introduce a bilateral filtering technique for signals on *any* domain admitting a diffusion operator. This filter coincides with the image bilateral in the planar case but can also be used to process signals on meshes, point clouds, and other domains with minimal modification. We can also process *geometric* signals such as xyz positions or mesh normals, enabling applications such as mesh smoothing. Our discretization is a faithful interpretation of the continuous formulation and naturally extends to a larger class of filtering tasks. More generally, our formulation builds upon and generalizes many previous image filtering ([4], [5]), mesh smoothing ([6], [7]), and distributional mode-finding ([8], [9]) techniques.

Iterative application of the bilateral leads to the *mean shift* filter, introduced in [10] and elsewhere, which has stronger denoising and edge-sharpening properties. We show that the standard formulation of the mean shift translates directly into our framework and can be used to filter signals like surface normals, which are naturally treated as signals with values on the sphere S^2. The result is a strong geometry filter illustrated in Figure 1.

Our method applies to several tasks from geometry processing including mesh smoothing, normal filtering on oriented point clouds, and curvature smoothing, all while respecting sharp edges. We also explore how modifications of our filter can be used to achieve interesting feature enhancement effects that respect sharp edges and prove that a slight modification of our method generates a smooth analog of a recently-introduced mesh vertex descriptor.

1.1 Contributions

The basic contribution of this paper is a framework for bilateral filtering of signals with *arbitrary* domain and

- *J. Solomon is with the Department of Computer Science, Stanford University, Stanford, CA, 94305.*
- *K. Crane is with the Department of Computer Science, Columbia University, New York, NY, 10027.*
- *A. Butscher is with the Max Planck Center for Visual Computing and Communication, Saarbrücken, Germany.*
- *C. Wojtan is with the Institute of Science and Technology Austria, Klosterneuburg, Austria.*
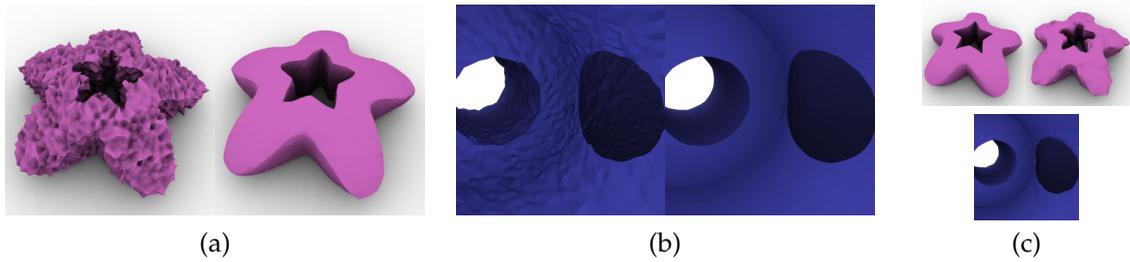
Fig. 1. (a,b) Examples of edge-preserving mesh smoothing using our mean-shift filter; noise is removed without mollifying sharp edges, and in (b) the circular holes are rounded; (c) comparisons with [1], [2], and [3], resp.

distance manifolds in Section 3. Section 4 develops schemes for mean-shift filtering using the generalized bilateral as a base, including proof that these methods are unconditionally convergent. We describe a stable, easy-to-implement, and convergent discretization in Section 5 and apply it to signals encountered in computer graphics in Sections 6 and 7, including geometric signals. Section 8 suggests additional applications and non-smoothing uses of our method.

2 BACKGROUND

[7], [11] survey work on mesh smoothing and fairing; we focus on bilateral geometry filtering schemes, which are the closest to our method.

2.1 Scalar Bilateral Filtering

The bilateral filter was introduced in [4] for filtering signals $f : I \to \mathbb{R}^n$ on an image I using a kernel that is the product of a spatial term W_s and an intensity term W_c:

$$\bar{f}(\mathbf{x}) = \frac{\int_I f(\mathbf{y}) W_s(\|\mathbf{x} - \mathbf{y}\|) W_c(\|f(\mathbf{x}) - f(\mathbf{y})\|) \, d\mathbf{y}}{\int_I W_s(\|\mathbf{x} - \mathbf{y}\|) W_c(\|f(\mathbf{x}) - f(\mathbf{y})\|) \, d\mathbf{y}} \quad (1)$$

Pixels are combined only when they are nearby both in space *and* in intensity. The *cross bilateral* filters a signal f_1 using intensity distances from another signal f_2 [12], [13]:

$$\bar{f}(\mathbf{x}) = \frac{\int_I f_1(\mathbf{y}) W_s(\|\mathbf{x} - \mathbf{y}\|) W_c(\|f_2(\mathbf{x}) - f_2(\mathbf{y})\|) \, d\mathbf{y}}{\int_I W_s(\|\mathbf{x} - \mathbf{y}\|) W_c(\|f_2(\mathbf{x}) - f_2(\mathbf{y})\|) \, d\mathbf{y}} \quad (2)$$

For instance, f_1 may be too noisy to have well-defined features, but it can instead be smoothed using features from f_2. Considerable work has been put into accelerating these filters; see [5], [14], [15] for recent examples.

Several methods apply bilateral filtering on non-image domains. Mostly, they map the domain to a regular grid and apply image processing methods; for instance, [16] uses the bilateral on a voxel grid for surface reconstruction. [14] 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 $\|\cdot\|_2$. [17] makes use of a bilateral on scalar mesh curvature signals, but their focus is on shape editing rather than evaluation of the bilateral itself.

2.2 Mesh Bilateral Filtering

One domain in which applications of the bilateral extend beyond grid-based methods is mesh fairing and smoothing. Table 1 lists several past approaches to extend the bilateral to mesh domains in this fashion. Despite the considerable amount of research devoted to mesh bilateral filtering, we find that *none* of the prior contributions exhibits the following desirable properties simultaneously, and most methods do not exhibit more than one at a time:

1) Use of intrinsic and smooth distance weights respecting the domain's metric without resorting to parameterization
2) Convergence in the limit of refinement or theory identifying the effects of the filter on an abstract surface
3) Applicability to multiple signal types and domains
4) Reduction to [4] for image signals

These desiderata characterize desirable behavior and convergence of generalized bilateral filtering techniques. For example, 1) ensures that the algorithm is tailored for mesh processing rather than adapting image-based strategies to local neighborhoods; avoiding local parameterization also contributes to algorithmic efficiency. Item 2) helps ensure that discretizations of filter integrals converge to their continuous counterparts; ad-hoc methods considering ring-based vertex neighborhoods on meshes do not satisfy this criterion. We include 3) to ensure that filters support multiple applications without tuning for a narrow set of domains, and 4) confirms our intuition that a filter is truly "bilateral" and thus can be understood using intuition from image processing. Our algorithm satisfies all these criteria and still performs comparably to the methods in Table 1.

2.3 Mean Shift Filtering

Mean shift filtering, introduced for image segmentation in [8], was shown to be equivalent to iterated cross bilateral filtering in [10]–before the bilateral filter formally was introduced. Given this connection, [29] and others make use of bilateral filter accelerations to accomplish mean shift. It produces strong feature-preserving denoising for images, but few attempts

Paper	Description
[6]	Bilaterally filters the height function of the surface over vertex tangent planes
[18]	Combines vertices with their projections onto nearby tangent planes; bilateral weights take into account distances to the tangent plane projection and to the tangent plane center
[19]	Uses bilateral filtering as part of a multi-pass approach to modify Laplacian smoothing using weights inspired by those in [6]
[20]	Iteratively applies a modification of [18] to improve surface normals for rendering.
[21]	Bilaterally filters jets on point clouds for reconstruction
[22]	Bilaterally filters mesh normals and then adjusts surface; weights are Gaussians in normal difference and an approximation of geodesic distance
[23]	Explicitly filters sharp edges and then faces separately using extrinsic distances, edge directions, normal difference, and projections as in [18]
[24]	Filters face normals using Euclidean distance between centroids and normal differences
[25]	Filters non-manifold surfaces by iteratively applying a bilateral similar to [18] and remeshing
[14]	Filters the difference between a mesh and its Laplace-smoothed counterpart in principal curvature coordinates using spin-images [26] for weights without a distance term
[3]	Denoises quadric surface approximations by extending [6]
[27]	Applies [24] with automatic parameter choice to normals and fits a new surface
[1]	Locally filters face normals using one-ring information; derives alternative implicit normal smoothing scheme using one-ring bilateral weights to change Laplacian operator
[28]	Approximates mesh bilateral filtering using separable filters along curvature directions

TABLE 1
A summary of previous attempts to adapt bilateral filtering to mesh domains.

have been made to apply it to mesh domains. [30] mean shifts mesh normals for segmentation; [31] proposes a mesh mean shift operator requiring local geodesic parameterizations. While attempts to mean shift signals *on* meshes or surfaces have been limited, mean shift filtering has been applied to different manifold-*valued* signals; for instance, [9], [32], [33] propose mean shift methods for filtering sphere-, analytic manifold-, and Riemannian manifold-valued signals, resp. Our framework bridges the gaps among a variety of existing methods in this domain.

3 GENERALIZED BILATERAL FILTERING

Take Σ to be the domain of a signal $f_1 : \Sigma \to \mathbb{R}^n$ equipped with a nonnegative symmetric kernel $K_\Sigma : \Sigma \times \Sigma \to \mathbb{R}$. Intuitively, we can think of $K_\Sigma(\mathbf{x}, \mathbf{y})$ as measuring the proximity between $\mathbf{x}$ and $\mathbf{y}$ on Σ. 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 Σ is any domain admitting a Laplacian operator L, such as a graph, surface, mesh, or point cloud, we can take K_Σ to be the kernel corresponding to a solution at some fixed $t > 0$ of the heat equation $\frac{\partial u}{\partial t} = Lu$, where $u(\mathbf{x}, t) : \Sigma \times [0, \infty) \to \mathbb{R}$; that is, $K_\Sigma(\mathbf{x}, \mathbf{y})$ measures how much a unit of heat diffuses from $\mathbf{x}$ to $\mathbf{y}$ along Σ in t time.

We can define a blurred version of f_1 as the convolution

$$\hat{f}_1(\mathbf{x}) = \frac{1}{z(\mathbf{x})} \int_\Sigma f_1(\mathbf{y}) K_\Sigma(\mathbf{x}, \mathbf{y}) \, d\mathbf{y} \qquad (3)$$

where $z(\mathbf{x})$ is the normalizing value $\int_\Sigma K_\Sigma(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}$. Let $\mathbf{T}(f)$ be the linear operator on square-integrable functions taking f_1 to $\hat{f}_1$; in other words, $\mathbf{T}$ blurs functions f with kernel K_Σ.

In parallel with the cross bilateral (2), take $f_2 : \Sigma \to \Gamma$ to be a function designed so that if $f_2(\mathbf{x})$ and $f_2(\mathbf{y})$

are distant, the signal f_1 at $\mathbf{x}$ and $\mathbf{y}$ should not be blended during filtering. We assume that Γ is a compact manifold with or without boundary; for instance, using RGB colors would yield $\Gamma = [0, 1]^3$, while using surface normals yields $\Gamma = S^2$, the unit sphere. We equip Γ with its own kernel $K_\Gamma : \Gamma \times \Gamma \to \mathbb{R}$.
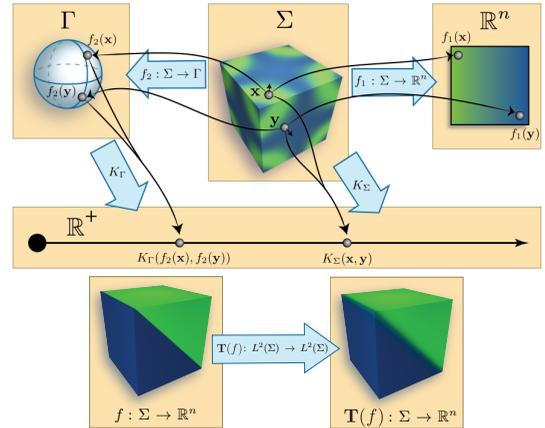
Fig. 2. Notation.

With this notation (illustrated in Figure 2) in place, we can introduce the *generalized cross-bilateral filter* as follows:

$$\bar{f}(\mathbf{x}) = \frac{\int_\Sigma f_1(\mathbf{y}) K_\Sigma(\mathbf{x}, \mathbf{y}) K_\Gamma(f_2(\mathbf{x}), f_2(\mathbf{y})) \, d\mathbf{y}}{\int_\Sigma K_\Sigma(\mathbf{x}, \mathbf{y}) K_\Gamma(f_2(\mathbf{x}), f_2(\mathbf{y})) \, d\mathbf{y}} \qquad (4)$$

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 $\|f_2(\mathbf{x}) - f_2(\mathbf{y})\|$.

We can re-express the cross bilateral using the diffusion operator $\mathbf{T}$ defined above. In particular, for fixed $\mathbf{p} \in \Gamma$ define numerator and denominator functions

as:

$$f_{\mathbf{p}}^{num}(\mathbf{y}) = f_1(\mathbf{y})K_\Gamma(\mathbf{p}, f_2(\mathbf{y})) \qquad (5)$$

$$f_{\mathbf{p}}^{den}(\mathbf{y}) = K_\Gamma(\mathbf{p}, f_2(\mathbf{y})) \qquad (6)$$

Then, we have

$$\bar{f}(\mathbf{x}) = \frac{\mathbf{T}[f_{f_2(\mathbf{x})}^{num}(\cdot)](\mathbf{x})}{\mathbf{T}[f_{f_2(\mathbf{x})}^{den}(\cdot)](\mathbf{x})} \qquad (7)$$

4 GENERALIZED MEAN SHIFT FILTERING

Bilateral filtering is reliable for minor denoising but is less effective on highly-noisy signals. In particular, the K_Γ term combines values only when they are similar; outliers thus will be influenced only slightly by their nearby counterparts. Furthermore, in certain scenarios it is desired not only to smooth signals but also to sharpen edges. For these purposes we propose a *generalized mean shift* filter below.

For fixed $\mathbf{x} \in \Sigma$, we can rewrite the denominator of the bilateral (4) as a probability distribution $h : \Gamma \to \mathbb{R}$ over Γ:

$$h_{\mathbf{x}}(\mathbf{p}) = \frac{1}{z(\mathbf{x})} \int_\Sigma K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(\mathbf{p}, f(\mathbf{y})) \, d\mathbf{y} \qquad (8)$$

where $z(\mathbf{x})$ is a normalizing constant so that $\int_\Gamma h_{\mathbf{x}}(\mathbf{p}) \, d\mathbf{p} = 1$. This function, constructed using the same technique as [34], represents the distribution of values of f near $\mathbf{x}$.

If $\Gamma = \mathbb{R}^n$ with $K_\Gamma(\mathbf{p}, \mathbf{q}) = e^{-\|\mathbf{p}-\mathbf{q}\|^2/\sigma^2}$, taking the gradient with respect to $\mathbf{p}$ we find that peaks $\mathbf{p}^*$ of $h_{\mathbf{x}}(\mathbf{p})$ satisfy

$$\mathbf{p}^* = \frac{\int_\Sigma f(\mathbf{y})K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(\mathbf{p}^*, f(\mathbf{y})) \, d\mathbf{y}}{\int_\Sigma K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(\mathbf{p}^*, f(\mathbf{y})) \, d\mathbf{y}} \qquad (9)$$

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

$$f^{(0)}(\mathbf{x}) = f(\mathbf{x}) \qquad (10)$$

$$f^{(k+1)}(\mathbf{x}) = \frac{\int_\Sigma f(\mathbf{y})K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(f^{(k)}(\mathbf{x}), f(\mathbf{y})) \, d\mathbf{y}}{\int_\Sigma K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(f^{(k)}(\mathbf{x}), f(\mathbf{y})) \, d\mathbf{y}} \qquad (11)$$

Each iteration applies a slightly modified cross bilateral (4). This scheme is an instance of the mean-shift filter [8], which converges unconditionally to peaks of $h_{\mathbf{x}}$ [35].

The derivation above assumes that $\Gamma = \mathbb{R}^n$. This restriction to $\mathbb{R}^n$ reflects a general drawback of bilateral filters and related integral operators, that they can take inputs on a manifold Γ but give outputs in the ambient $\mathbb{R}^n$; we are unaware of a bilateral filter that does not have this property without postprocessing. In particular, filters including [1] modify surface normals (on the sphere S^2) but result in filtered versions without unit length; these filters can be difficult to understand and control. The description of the mean shift as a mode-finding technique, however, is valid for any Γ independent of its embedding, and we can

take advantage of this observation to build denoising methods that are *intrinsic* to Γ.

More formally, our construction of h remains valid when $\Gamma \neq \mathbb{R}^n$. For instance, we can equip $\Gamma = S^2$ with the Von Mises–Fisher kernel $K_\Gamma(\mathbf{p}, \mathbf{q}) = e^{\mathbf{p} \cdot \mathbf{q}/\sigma}$ for unit vectors $\mathbf{p}$ and $\mathbf{q}$, used to represent isotropic distributions on the unit sphere [36]. In this case, a similar argument to the one above yields the mean-shift iteration:

$$f^{(0)}(\mathbf{x}) = f(\mathbf{x}) \qquad (12)$$

$$f^{(k+1)}(\mathbf{x}) = \frac{\int_\Sigma f(\mathbf{y})K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(f^{(k)}(\mathbf{x}), f(\mathbf{y})) \, d\mathbf{y}}{\|\int_\Sigma f(\mathbf{y})K_\Sigma(\mathbf{x}, \mathbf{y})K_\Gamma(f^{(k)}(\mathbf{x}), f(\mathbf{y})) \, d\mathbf{y}\|} \qquad (13)$$

Each iterate has unit length and thus remains on S^2. This new iterative scheme is an instance of the spherical mean shift algorithm in [9] being carried out in parallel at each $\mathbf{x} \in \Sigma$, proving its convergence and its qualitative similarity to the Euclidean case. Iterations of (13) are effectively averaging unit vectors; while this is the mathematically correct operation to carry out according to the Von Mises–Fisher kernel, there is some potential for numerical instability when σ is large. We have not observed such issues in the applications we propose for reasonable choices of σ; particular values are documented in the supplementary material.

We have concentrated above on two simple domains Γ: subsets of $\mathbb{R}^n$ and the sphere S^2. These are by no means the only choices of Γ that yield convergent mode-finding schemes. [32] and [33] provide mean shift methods when data is on analytic or Riemannian manifolds, resp., that can be adapted to our framework on Σ in a similar manner.

5 DISCRETIZATION

We employ a signal processing technique similar to that in [5] to evaluate the bilateral filter on discrete domains Σ (Algorithm 1). Our method applies essentially the same computations to $f_{\mathbf{p}}^{den}$ as $f_{\mathbf{p}}^{num}$, so for ease of notation during its development denote $f_{\mathbf{p}}$ as one of $f_{\mathbf{p}}^{num}$ or $f_{\mathbf{p}}^{den}$.

Suppose that we choose samples $\mathbf{p}_1, \ldots, \mathbf{p}_m \in \Gamma$ and a corresponding partition of unity $\phi_1, \ldots, \phi_m : \Gamma \to \mathbb{R}$ such that a function $g : \Gamma \to \mathbb{R}$ can be approximated as $g(\mathbf{p}) \approx \sum_i g(\mathbf{p}_i)\phi_i(\mathbf{p})$. Note that under mild continuity and compactness conditions, we can construct sequences of partitions such that the approximation converges to $g(\mathbf{p})$ as $m \to \infty$. This discretization is similar to the use of finite element bases to express functions on surfaces [37]; for instance, on a triangle mesh, piecewise linear "hat" functions can serve as an appropriate partition of unity.

Define $g_i(\mathbf{x}) = f_{\mathbf{p}_i}(\mathbf{x})$; this function can be computed for all x in Σ by evaluating f_1 and K_Γ as in (5) and (6). The blurring operation (3) is then applied to obtain $\hat{g}_i(\mathbf{x}) = \mathbf{T}[g_i](\mathbf{x})$. For instance, if Σ is an

---

Input : Signal to be filtered $f_1 : \Sigma \to \mathbb{R}^n$
          Cross bilateral function $f_2 : \Sigma \to \Gamma$
          Samples $\mathbf{p}_1, \ldots, \mathbf{p}_m \in \Gamma$
          Partition of unity $\phi_1, \ldots, \phi_m : \Gamma \to \mathbb{R}$
Output: Filtered signal $\bar{f} : \Sigma \to \mathbb{R}^n$

$\bar{f}^{num}(\mathbf{x}), \bar{f}^{den}(\mathbf{x}) \leftarrow 0 \, \forall \mathbf{x} \in \Sigma;$ *Initialization*
for $i = 1$ *to* m **do**
    $g^{num}(\mathbf{x}) \leftarrow f_1(\mathbf{x}) K_\Gamma(f_2(\mathbf{x}), \mathbf{p}_i);$ *Weight signals*
    $g^{den}(\mathbf{x}) \leftarrow K_\Gamma(f_2(\mathbf{x}), \mathbf{p}_i);$
    $\hat{g}^{num}(\mathbf{x}) \leftarrow \mathbf{T}[g^{num}](\mathbf{x});$ *Apply blur operator*
    $\hat{g}^{den}(\mathbf{x}) \leftarrow \mathbf{T}[g^{den}](\mathbf{x});$
    $\bar{f}^{num}(\mathbf{x}) \leftarrow \bar{f}^{num}(\mathbf{x})$ *Collect*
               $+ \hat{g}^{num}(\mathbf{x}) \phi_i(f_2(\mathbf{x}));$
    $\bar{f}^{den}(\mathbf{x}) \leftarrow \bar{f}^{den}(\mathbf{x}) + \hat{g}^{den}(\mathbf{x}) \phi_i(f_2(\mathbf{x}));$
end
$\bar{f}(\mathbf{x}) \leftarrow \bar{f}^{num}(\mathbf{x})/\bar{f}^{den}(\mathbf{x});$ *Normalize*

Algorithm 1: Generalized bilateral filtering algorithm

image then $\mathbf{T}$ will be a Gaussian blur, while mesh bilateral filters would implement $\mathbf{T}$ using diffusion. Our bilateral filter is thus approximated as:

$$\bar{f}(\mathbf{x}) \approx \frac{\sum_i \hat{g}_i^{num}(\mathbf{x}) \phi_i(f_2(\mathbf{x}))}{\sum_i \hat{g}_i^{den}(\mathbf{x}) \phi_i(f_2(\mathbf{x}))} \qquad (14)$$

We show several concrete applications of bilateral filtering simply by applying this formulation to various domains and kernels. If K_Γ is straightforward to evaluate, the only time-consuming step is generating the functions $\hat{g}_i$ from g_i; that is, the time complexity of this algorithm is essentially that of carrying out $2m$ blurs (3).

6 PROCESSING SCALAR SIGNALS

Before introducing novel domains and signals, we verify that our bilateral filter applied to grayscale images reduces to the one presented in [5]. 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 $K_\Gamma(p, q) = W_c(|p - q|)$. 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 $\phi_i : [0, 1] \to \mathbb{R}$ to be the piecewise linear hat function centered at p_i with width $2/m{-}1$. Then, (14) coincides with the "signal processing approximation" in [5]. The approximation is indistinguishable from the exact bilateral on most images for m as low as 20, and it can be carried out using down/up-sampling or methods like [38], [39] for $\mathbf{T}$ in (3).

Generalizing somewhat, suppose we take Σ to be a mesh with vertices V, edges E, and triangular faces F. We represent scalar functions on Σ as vectors $\mathbf{v} \in \mathbb{R}^{|V|}$ and construct a "cotangent Laplacian" matrix

$L \in \mathbb{R}^{|V| \times |V|}$ with diagonal mass matrix $A \in \mathbb{R}^{|V| \times |V|}$ imitating the Laplacian operator on the smooth surface approximated by Σ [40]. We compute $\mathbf{T}(\mathbf{v})$ using heat flow using a single implicit time step $\mathbf{T}(\mathbf{v}) \approx (I + \Delta t A^{-1} L)^{-1} \mathbf{v}$. Multiple time steps or a higher-order discretization yield closer approximations, but the damping effect of a single implicit step has few perceptual differences and is faster to carry out; furthermore, it can be viewed as an isotropic instance of the screened Poisson equation [41], which may suggest future research directions making bilateral filtering faster or more anisotropic. Since we apply $\mathbf{T}$ several times, we pre-factor time time step matrix using the sparse LU method in [42]. We keep $\Gamma = [0, 1]$ with Gaussian kernel $K_\Gamma(x, y) = e^{-|x-y|^2/\sigma^2}$.
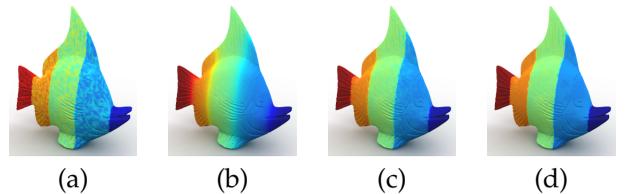
Fig. 3. A noisy function (a) smoothed using Laplacian diffusion (b), the generalized bilateral, (c), and the mean shift (d). Diffusion does not preserve signal edges, the bilateral removes most of the noise while preserving edges, and the mean shift provides strong denoising.

If we take $f_1 = f_2 \equiv f : \Sigma \to \mathbb{R}$, the generalized bilateral blurs f while preserving its discontinuities. Figure 3 shows the output of this method and the iterative mean shift on a noisy texture. Unlike the image bilateral and mesh methods relying on planar projection or parameterization, this bilateral respects the metric of Σ regardless of the width of K_Σ.
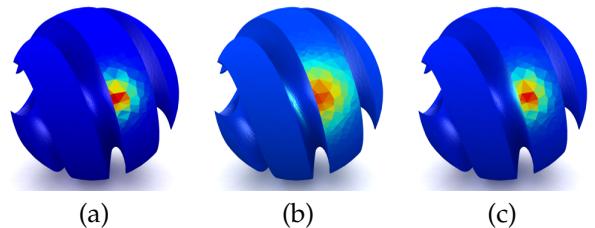
Fig. 4. Kernel of the normal cross bilateral (a). Increasing the reach of K_Σ widens the kernel (b), while increasing that of K_Γ allows the kernel to continue over sharp edges (c).

7 MESH DENOISING

We can extend the method in Section 6 by considering cross bilaterals for which Γ is not $[0, 1]$. Most importantly, suppose $\Gamma = S^2$, the unit sphere, and take f_2 to be the signal $\mathbf{N} : F \to S^2$ given by unit face normals. Our signal now is on mesh *faces* rather than vertices

to avoid ambiguous normals along sharp edges. So, we replace L from Section 6 with the dual 0-form Laplacian $d \star d\star$ from discrete exterior calculus [43]. Figure 4 illustrates the bilateral kernel $K_\Sigma K_\Gamma$ in this context.
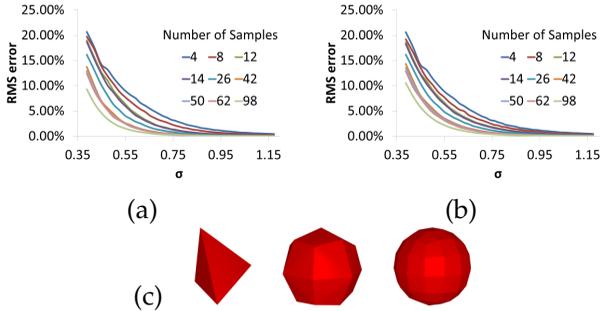
Fig. 6. Mean-square reconstruction error of Von Mises–Fisher kernels of assorted sizes σ using varying numbers of sample points and (a) piecewise-linear interpolation or (b) meshless interpolation; (c) approximations of the unit sphere for (a) with 4, 26, and 98 samples, resp.

A partition of unity on S^2 is obtained using a regular polyhedron inscribed within S^2; each ϕ_i corresponds to a piecewise linear hat function centered at a vertex of the polyhedron projected to S^2. An alternative more efficient and smoother partition of unity paralleling meshless integration is to use Von Mises–Fisher kernels centered at sample points on the unit sphere normalized to sum to 1; we choose the width of the kernels to be half the average distance from each sample to its closest neighbor. We find little qualitative difference between these approaches and show experiments determining sufficient sampling rates for different kernel sizes in Figure 6. Applications of this filter to scalar functions on Σ are shown in Figure 5; values are not combined over sharp edges since the normal $\mathbf{N}$ has a discontinuity there.

If we filter $\mathbf{N} : \Sigma \to S^2$ itself, we obtain a denoised normal field over Σ; this step evaluates the normal vector bilateral proposed in [1], although their method resorts to a somewhat severe approximation effective for small blending radii. As in [1] and others, we subsequently adjust Σ to match the denoised normals using the method in [7]. While [7] is presented in discrete terms, it simply is solving a Poisson-type equation to recover a nearby surface with the adjusted normals; it is designed not to induce shrinkage and other artifacts. Figure 7 compares denoising results of the normal bilateral and mean shift filters with those of some previous methods; of course, the choice of reconstruction methods is independent of our filter and can be replaced if desired.

8 ADDITIONAL APPLICATIONS

Here we provide some applications of our method outside of mesh processing. These show its broad
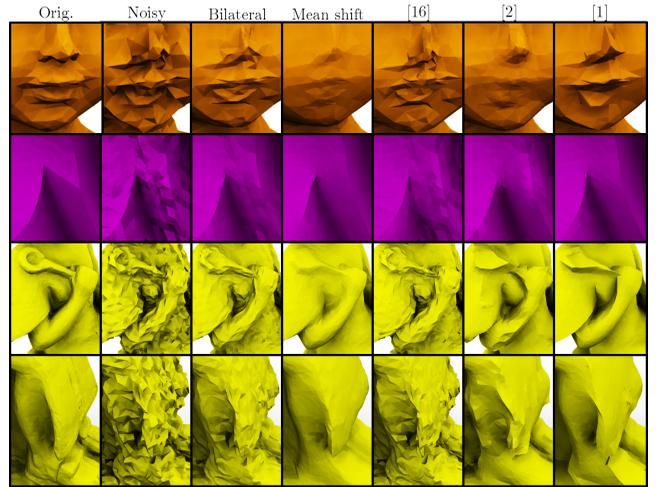
Fig. 8. Closer views of some examples from Figure 7 with increased contrast (Figure 7 is rendered with per-face Lambertian shading for simplicity).

variety of applications for smoothing and other signal processing tasks.

8.1 Oriented Point Clouds

Algorithms like [45] for surface reconstruction rely on *oriented* point clouds, which contain both sample points and their normals, to generate meshes; the normals help decipher tangent directions, orientation, and connectivity. 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 *a posteriori*.

Fortunately, [46] introduces a Laplacian for signals on point clouds with provable convergence. Laplacian heat diffusion along with the bilateral term ensures that edges are preserved and that surface topology is respected while combining "nearby" normals. Figure 9 shows examples of reconstruction using [45] with and without bilateral normal filtering on point clouds from [47].

8.2 Bilateral and Mean Shift on Other Signals

The filters we discuss above are by no means the only ones that fit in our framework. Additional domains and signals to which we could apply Algorithm 1 include:

- Textures equipped with a blurring operator from MIP maps or a Laplacian pulled back from the mesh
- Signals on polygonal meshes using the Laplacian from [48] for diffusion
- Point clouds with skeletons as in [49], so points are combined when they are close on the skeleton and with respect to point cloud Laplacian heat flow
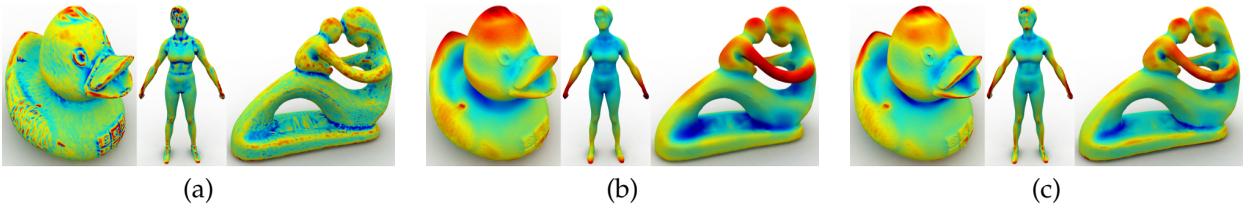
Fig. 5. Noisy mean curvature values obtained from a one-ring computation (a), Laplacian diffusion-smoothed mean curvatures (b), and bilateral-filtered mean curvatures (c).
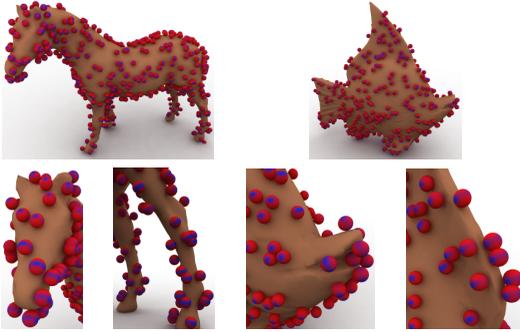
Fig. 10. Local normal histograms describing the distribution of normals near a given face, color-coded on the unit sphere; histograms are shown at a random set of faces.

- Quadric surface approximations as in some works in Table 1, with cross bilateral signals suggested here or in the original papers
- Graphs with discrete Laplacian diffusion
- Range images with RGB or normals for the cross bilateral
- Volumetric signals with heat flow using f_1 as a density
- Simplicial complexes with combinatorial Laplacian flow

Many of these applications are outside 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 below.

8.3 Local Histograms

[34] suggests that the histogram $h_{\mathbf{x}}(\mathbf{p})$ in (8) has value for understanding signals on images; in particular, they use this function to understand the distribution of intensities in some smoothly-weighted neighborhood of each pixel. An *identical* formulation applies to our more general setting. In particular, evaluation of $h(\mathbf{p}_i; \mathbf{x}) \ \forall \mathbf{x} \in \Sigma$ occurs while computing the samples in the denominator in Algorithm 1. Thus, we can efficiently extract local histograms of signals $f : \Sigma \to \Gamma$ using the same partition of unity approach. This allows for the direct evaluation of the filters in [34] 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. For example, once again taking $f : \Sigma \to \Gamma = S^2$ to be the normal vector signal on Σ, the histogram $h_{\mathbf{x}}(\mathbf{p})$ at a fixed $\mathbf{x} \in \Sigma$ now represents the distribution over S^2 of normal vectors to Σ near $\mathbf{x}$. This distribution can be viewed (after suitable rotation) as a version of the SHOT descriptor introduced in [50] with smoothly varying, intrinsic heat kernel weights on Σ rather than extrinsic distance weights, with straightforward regularization control by changing blurring radii on Σ and Γ. Figure 10 shows some examples of normal vector histograms computed using this technique. These images show that our histograms of normals are equally informative to the SHOT descriptor; viewed as probability distributions on the unit sphere, these histograms also suggest the possibility of applying filtering techniques such as [51] to meshed domains.

8.4 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_Σ and K_Γ.

The most obvious potential change in K_Σ or K_Γ might be in the choice of smoothing kernels. We implicitly have made use of this flexibility 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 a member 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 containing kernel values $K_\Sigma : \mathbb{R}^{|V|} \times \mathbb{R}^{|V|} \to \mathbb{R}$; the theory of Schwartz kernels can be used to prove a similar statement in the continuous limit [52]. The bilateral simply reweights these linear kernels to respect signal edges.

Although fully exploring the domain of feature-preserving mesh operations is worthy of a larger study, Figure 11 shows examples of the application of

(a)   (b) 0.296(d) 0.164(f) 0.212(g) 0.241   (b) 0.114(c) 0.109(d) 0.104   (e)   (f) 0.105(g) 0.110

(a)   (b) 0.111   (d) 0.074   (e)   (f) 0.094   (g) 0.087

(a)   (b) 0.022 (c) 0.019 (d) 0.013   (e)   (f) 0.021 (g) 0.012

(a)   (b) 0.243 (c) 0.156 (d) 0.109   (e)   (f) 0.171 (g) 0.120

(a)   (b) 0.257 (c) 0.060 (d) 0.069 (g) 0.236   (a)   (d)

(a)   (b) 0.281   (c) 0.174   (d) 0.121   (e)   (g) 0.166

(a)   (b) 0.120(c) 0.119(d) 0.111   (e)   (f) 0.163(g) 0.139   (b) 0.318(d) 0.174(f) 0.213(g) 0.232

(a)   (b) 0.024   (d) 0.011   (h) 0.010   (a)   (b) 0.163(d) 0.087   (h)
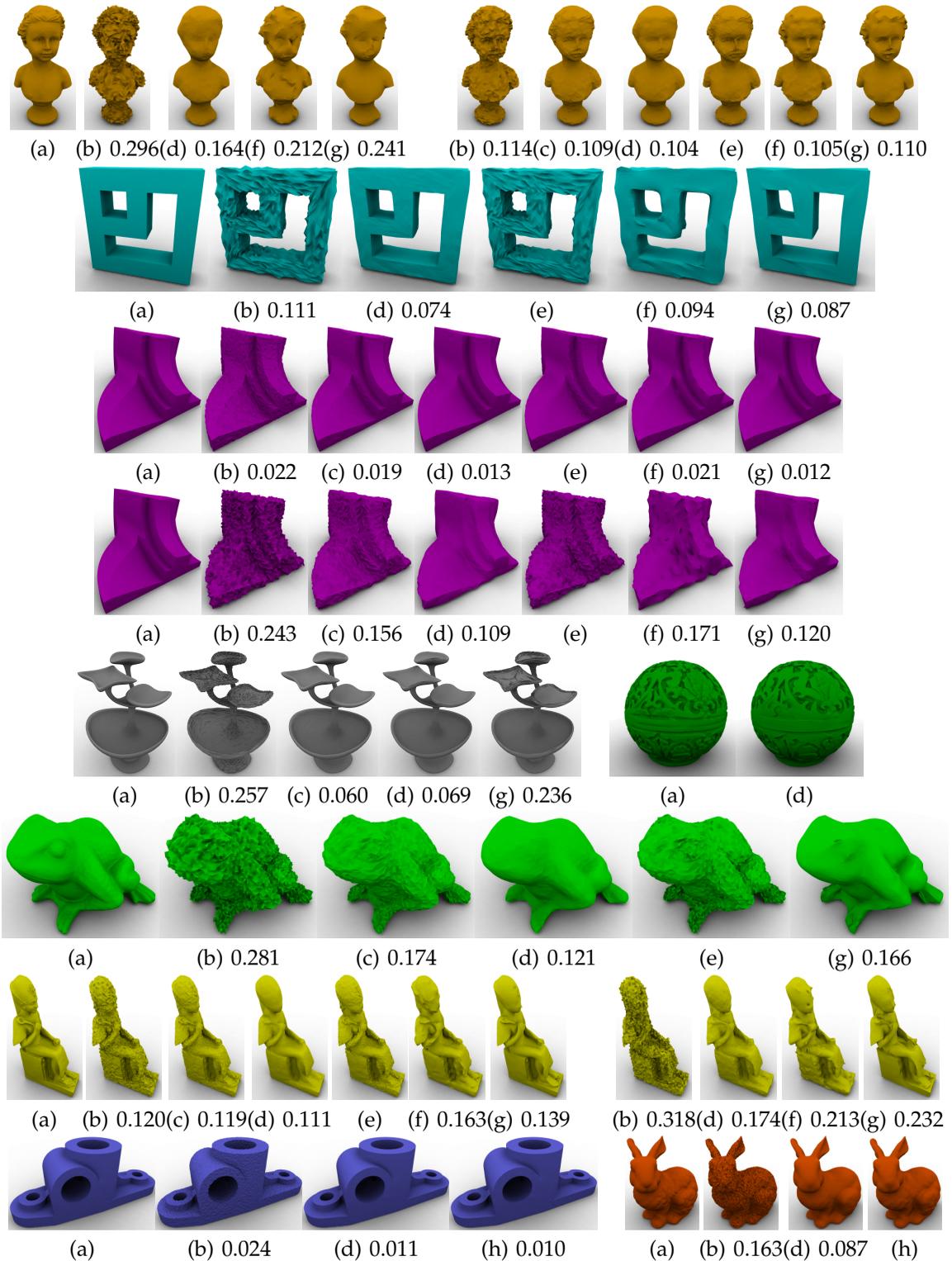
Fig. 7. Noise is added to (a) to generate test case (b). We smooth using our bilateral (c) and mean shift (d) filters and provide comparisons with [18] (e), [2] (f), [1] (g), and [3] (h). Perceptual STED distance [44] from original non-noisy surface is shown underneath when computable and relevant. Noise is generated by randomly displacing mesh vertices under a uniform distribution, except for the bottom row, which uses tests from [3]. Data for Figure 9 is from real-world scans; here we opt to generate synthetic noise to enable use of the STED metric.
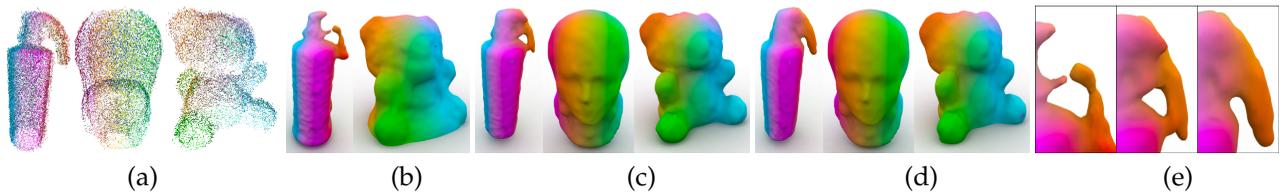
Fig. 9. Surface reconstruction from the oriented point cloud (a; rendered using normal vectors for lighting with hue chosen by position), with original normals (b; bust case fails), bilaterally-smoothed normals (c), and mean-shifted normals (d). Bilateral and mean shift filtering create considerably better reconstruction results; even in the difficult case of the fire extinguisher cloud, mean shift filter is able to generate normals that separate the handle from the body of the extinguisher (e).
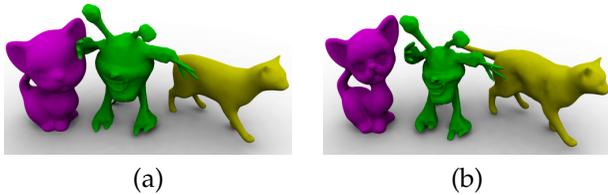
Fig. 11. Examples of non-blurring bilateral filters to achieve interesting edge-preserving shape deformations.

our bilateral where the kernel K_Σ has been replaced with the kernels of other linear operators. In particular, we use the unsharp mask for K_Σ while keeping K_Γ Gaussian in mesh normals. The resulting filter is applied to mean curvature normals, yielding meshes with exaggerated curvature while avoiding artifacts like ringing near sharp corners.

9 DISCUSSION

We have written an implementation of our algorithm in C++, taking advantage of templates to encode Algorithm 1 in full generality; we use OpenMP directives to achieve parallel evaluation of the blurs needed for each sample $\mathbf{p}_i$. On a four-core 2.40 GHz Intel Xeon machine, this naïve implementation can apply bilateral filters to mesh normals on 12946 faces in 2.72 seconds using 42 sample points on S^2. Subsequent iterations for the mean shift are even faster, since they can reuse the same prefactored heat flow matrix; this method converges in as few as five to ten iterations.

Faster run times could be achieved with an optimized implementation and faster linear solvers. Our runtime is limited by the time it takes to blur $2m$ signals using K_Σ, so fewer samples $\mathbf{p}_i \in S^2$ make for better timings; we can cut our number of samples to half of the ones listed here with reasonable effect but slight visible artifact in exchange for a faster filter.

Figure 7 compares against recent work on mesh smoothing; larger image of representative examples are shown in Figure 8. We apply uniform noise of varying sizes to mesh vertices and then apply our and other smoothing methods to recover the original shape. We show the perceptual "STED" distance between the filtered signals and the original [44], [53]. In general, we find that our algorithm behaves comparably with state-of-the-art, yielding small STED distances to the original meshes even when compared to the results of more specialized papers.

9.1 Limitations

While the theoretical and practical properties of our generalized bilateral filter make it an obvious choice in a variety of circumstances, it is important to note tasks for which our construction is not as well-suited. In particular, we require Γ to be compact (possibly with boundary) and to admit a partition of unity; this assumption is fairly weak for signals such as mesh normals, which live on S^2, but makes it difficult to consider signals like the tangent plane projections in [18] that can take values within a large part of $\mathbb{R}^n$.

One property exhibited by mesh smoothing algorithms making use of geometric flows rather than integral operators like the bilateral is that they somehow "directly" filter the geometry rather than treating it as a signal. In fact, our method as-is actually can deal with geometry in at least two ways. First, as proposed in Section 7, we can use normals to process geometry indirectly. This approach has the advantage that edges in the geometry become discontinuities in the signal, whereas xyz positions on a mesh are continuous everywhere. Given the reconstruction method in [7], one can view the normal signal as an alternative non-Euclidean expression of geometry that can be processed like any other embedding. Second, our bilateral could be applied directly to xyz positions as the signal on Σ using normals on $\Gamma = S^2$. This alternative better mimics flows, but we found it less effective than normal processing and omitted the results. Normal processing has been shown repeatedly to be a highly-effective denoising technique, so we are hardly the first to come to this conclusion [24], [7], [1]. We leave the interpretation of our filter as an anisotropic flow as in [54] for images for future research.

A related issue that will require additional study is the effect of the reconstruction in [7] on the convergence properties of our normal-based mesh processing technique. Nonetheless, consistency for signals

on fixed irregular domains is a valuable feature of our method, and one that is not guaranteed by any existing method.

10 CONCLUSION

The sheer number of attempts to discretize bilateral filtering on non-image domains illustrated in Table 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. Feature-preserving filters can be achieved on arbitrary domains simply by choosing domains Σ, Γ and kernels K_Σ, K_Γ, with the assumption that Γ 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 Γ and the time it takes to apply K_Σ, the latter of which often boils 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 to its application in other settings. For instance, in image processing, some results show that distances between signatures for commonly-used cross bilateral signals may not be measured using the Euclidean metric but rather along some underlying manifold [55], [56]; this type of relationship can be encoded in our framework by defining Σ to be a part of the image plane and Γ 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 they begin to reveal the exciting potential implications of a reliable generalized bilateral filtering technique.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral Normal Filtering for Mesh Denoising," *IEEE Trans. Vis. Comp. Graph.*, vol. 17, no. 10, pp. 1521–1530, 2011.

[2] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Comp. Graph. Forum*, vol. 23, pp. 391–400, 2004.

[3] H. Fan, Y. Yu, and Q. Peng, "Robust Feature-Preserving Mesh Denoising Based on Consistent Subneighborhoods," *IEEE Trans. Vis. Comp. Graph.*, vol. 16, no. 2, pp. 312–324, 2010.

[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Int. Conf. Comp. Vis.*, 1998, pp. 839–846.

[5] S. Paris and F. Durand, "A Fast Approximation of the Bilateral Filter using a Signal Processing Approach," in *Proc. European Conf. Comp. Vis.*, 2006, pp. 568–580.

[6] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, pp. 950–953, 2003.

[7] X. Sun, P. Rosin, R. Martin, and F. Langbein, "Fast and Effective Feature-Preserving Mesh Denoising," *IEEE Trans. Vis. Comp. Graph.*, vol. 13, no. 5, pp. 925–938, 2007.

[8] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. & Mach. Intel.*, vol. 24, no. 5, pp. 603–619, 2002.

[9] T. Kobayashi and N. Otsu, "Von Mises-Fisher Mean Shift for Clustering on a Hypersphere," in *Int. Conf. Pattern Rec.*, 2010, pp. 2130–2133.

[10] J. Van de Weijer and R. Van den Boomgaard, "Local mode filtering," in *Comp. Vis. and Pattern Rec.*, vol. 2, 2001, pp. 428–433.

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

[12] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.*, vol. 23, pp. 664–672, 2004.

[13] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 673–678, Aug. 2004.

[14] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian KD-trees for fast high-dimensional filtering," *ACM Trans. Graph.*, vol. 28, pp. 21:1–21:12, 2009.

[15] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," in *Comp. Graph. Forum*, vol. 29, 2010, pp. 753–762.

[16] A. Miropolsky and A. Fischer, "Reconstruction with 3D geometric bilateral filter," in *Symp. Solid Modeling and App.* Eurographics Association, 2004, pp. 225–229.

[17] M. Eigensatz, R. W. Sumner, and M. Pauly, "Curvature-Domain Shape Processing," *Comp. Graph. Forum*, vol. 27, no. 2, pp. 241–250, 2008.

[18] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, pp. 943–949, 2003.

[19] G.-F. Hu, Q.-S. Peng, and A. R. Forrest, "Robust mesh smoothing," *J. Comp. Sci. Technol.*, vol. 19, pp. 521–528, 2004.

[20] T. Jones, F. Durand, and M. Zwicker, "Normal improvement for point rendering," *Comp. Graph. and App.*, vol. 24, no. 4, pp. 53–56, 2004.

[21] F. Duguet, F. Durand, and G. Dettrakis, "Robust Higher-Order Filtering of Points," INRIA, Tech. Rep. RR-5165, Apr. 2004.

[22] Q. Hou, L. Bai, and Y. Wang, "Mesh Smoothing via Adaptive Bilateral Filtering," in *ICCS*, vol. 3515, 2005, pp. 273–280.

[23] T. Shimizu, H. Date, S. Kanai, and T. Kishinami, "A New Bilateral Mesh Smoothing Method by Recognizing Features," in *Proc. Comp. Aided Design and Comp. Graph.*, 2005, pp. 281–286.

[24] K.-W. Lee and W.-P. Wang, "Feature-Preserving Mesh Denoising via Bilateral Normal Filtering," in *Proc. Comp. Aided Design and Comp. Graph*, 2005, pp. 275–280.

[25] C. Wang, "Bilateral recovering of sharp edges on feature-insensitive sampled meshes," *IEEE Trans. Vis. Comp. Graph.*, vol. 12, no. 4, pp. 629–639, 2006.

[26] A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Anal. & Mach. Intel.*, vol. 21, pp. 433–449, 1999.

[27] M. Nociar and A. Ferko, "Feature-preserving mesh denoising via attenuated bilateral normal filtering and quadrics," in *Proc. Spring Conf. on Comp. Graph.*, 2010, pp. 149–156.

[28] G. Vialaneix and T. Boubekeur, "SBL Mesh Filter: A Fast Separable Approximation of Bilateral Mesh Filtering," in *Vision, Modeling and Visualization*, 2011.

[29] S. Paris and F. Durand, "A Topological Approach to Hierarchical Segmentation using Mean Shift," in *Comp. Vis. and Pattern Rec.*, June 2007, pp. 1–8.

[30] H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Feature Sensitive Mesh Segmentation with Mean Shift," in *Proc. Shape Modeling and App.*, 2005, pp. 238–245.

[31] A. Shamir, L. Shapira, and D. Cohen-Or, "Mesh analysis using geodesic mean-shift," *Vis. Comp.*, vol. 22, pp. 99–108, 2006.

[32] R. Subbarao and P. Meer, "Nonlinear Mean Shift for Clustering over Analytic Manifolds," in *Comp. Vis. and Pattern Rec.*, 2006, pp. 1168–1175.

[33] ——, "Nonlinear Mean Shift over Riemannian Manifolds," *Int. J. Comp. Vis.*, vol. 84, pp. 1–20, August 2009.

[34] M. Kass and J. Solomon, "Smoothed Local Histogram Filters," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 100:1–100:10, 2010.

[35] X. Li, Z. Hu, and F. Wu, "A note on the convergence of the mean shift," *Pattern Recognition*, vol. 40, no. 6, pp. 1756–1762, 2007.

[36] R. Fisher, "Dispersion on a Sphere," *Proc. Royal Society of London*, vol. 217, no. 1130, pp. 295–305, 1953.

[37] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover, 2012.

[38] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Comm.*, vol. 31, no. 4, pp. 532–540, 1983.

[39] R. Deriche, "Recursively Implementing the Gaussian and its Derivatives," INRIA, Tech. Rep. 1893, 1993.

[40] R. MacNeal, "The solution of partial differential equations by means of electrical networks," Ph.D. dissertation, Caltech, 1949.

[41] M. Chuang and M. Kazhdan, "Interactive and anisotropic geometry processing using the screened Poisson equation," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 57:1–57:10, Jul. 2011.

[42] T. Davis, "Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol. 30, pp. 196–199, June 2004.

[43] A. Hirani, "Discrete Exterior Calculus," Ph.D. dissertation, Caltech, 2003.

[44] L. Váša and V. Skala, "A Perception Correlated Comparison Method for Dynamic Meshes," *IEEE Trans. Vis. Comp. Graph.*, vol. 17, no. 2, pp. 220–230, Feb. 2011.

[45] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," in *Proc. Symp. Geom. Proc.*, 2006, pp. 61–70.

[46] M. Belkin, J. Sun, and Y. Wang, "Constructing Laplace Operator from Point Clouds in $\mathbb{R}^d$," in *Proc. Symp. Disc. Alg.*, 2009, pp. 1031–1040.

[47] M. Reinhardt, S. Dach, M. Backasch, H. Ben Amor, B. Jung, C. Schlegel, and T. Schröder. (2011, September) 3D Scan 2.0. [Online]. Available: http://vr.tu-freiberg.de/scivi/

[48] M. Alexa and M. Wardetzky, "Discrete Laplacians on general polygonal meshes," *ACM Trans. Graph.*, vol. 30, pp. 102:1–102:10, Aug. 2011.

[49] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point Cloud Skeletons via Laplacian Based Contraction," in *Shape Modeling Int.*, June 2010, pp. 187–197.

[50] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. European Conf. Comp. Vis.*, 2010, pp. 356–369.

[51] M. Hadwiger, R. Sicat, J. Beyer, J. Krüger, and T. Möller, "Sparse PDF maps for non-linear multi-resolution image operations," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 133:1–133:12, Nov. 2012.

[52] L. Hörmander, *The analysis of linear partial differential operators: Distribution theory and Fourier analysis*. Springer-Verlag, 1990.

[53] L. Váša and O. Petřík, "Optimising Perceived Distortion in Lossy Encoding of Dynamic Meshes," *Comp. Graph. Forum*, vol. 30, no. 5, pp. 1439–1449, 2011.

[54] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. & Mach. Intel.*, vol. 24, no. 6, pp. 844–847, jun 2002.

[55] G. Carlsson, T. Ishkhanov, V. Silva, and A. Zomorodian, "On the Local Behavior of Spaces of Natural Images," *Int. J. Comput. Vision*, vol. 76, pp. 1–12, January 2008.

[56] G. Peyré, "Manifold models for signals and images," *Comp. Vis. and Image Understanding*, vol. 113, no. 2, pp. 249–260, Feb. 2009.

Justin Solomon is a PhD candidate in the Geometric Computing Group of Stanford University's Department of Computer Science. He also received a BS in Mathematics and Computer Science (2010) and an MS in Computer Science (2012) at Stanford. His areas of study include geometry processing, computer graphics, and numerical methods with a focus on understanding geometric data. He is supported by the Hertz, NDSEG, and NSF graduate fellowships.

Keenan Crane is a PhD student in the Department of Computing and Mathematical Sciences at Caltech. He received a BS in Computer Science from the University of Illinois at Urbana Champaign in 2006. He is the recipient of a Google PhD Fellowship and a National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship. His current research focuses on discrete differential geometry with applications in digital geometry processing.

Adrian Butscher is a senior research scientist with the Max Planck Institute for Computer Science. He received his PhD in mathematics at Stanford University in 2000. His current research interests include discrete and continuous differential geometry with applications in digital geometry processing.

Chris Wojtan received his B.S. in Computer Science in 2004 from the University of Illinois in Urbana Champaign and his Ph.D. in Computer Graphics from the Georgia Institute of Technology in 2010. He was awarded a National Science Foundation Graduate Research Fellowship, the Georgia Tech Sigma Xi Best Ph.D. Thesis Award, and the Microsoft Visual Computing Award. Chris is currently an Assistant Professor at the Institute of Science and Technology Austria (IST Austria), and his research interests are physically-based animation and geometry processing.