Feature Preserving Smoothing of 3D Surface Scans

by

Thouis Raymond Jones

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[February 2004]
September 2003

Author .
Department of Electrical Engineering and Computer Science
September 16, 2003

Certified by.. .....
Frédo Durand
Assistant Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Feature Preserving Smoothing of 3D Surface Scans

by

Thouis Raymond Jones

Submitted to the Department of Electrical Engineering and Computer Science
on September 16, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

With the increasing use of geometry scanners to create 3D models, there is a rising need for effective denoising of data captured with these devices. This thesis presents new methods for smoothing scanned data, based on extensions of the bilateral filter to 3D. The bilateral filter is a non-linear, edge-preserving image filter; its extension to 3D leads to an efficient, feature preserving filter for a wide class of surface representations, including points and "polygon soups."

Thesis Supervisor: Frédo Durand
Title: Assistant Professor

Contents

1 Introduction **5**

1.1 Our Goal 7

1.2 Approach 7

2 Related Work **9**

2.1 Smoothing Without Preserving Features 9

2.2 Anisotropic Diffusion 10

2.3 Wiener Filtering 12

2.4 The Bilateral Filter 13

2.5 Surface Reconstruction Methods 14

2.6 Discussion 15

3 Robust Estimation, Anisotropic Diffusion and the Bilateral Filter **16**

3.1 Robust Estimation 16

3.2 Anisotropic Diffusion 20

3.3 Anisotropic Diffusion and Robust Estimation 23

3.4 The Bilateral Filter 26

4 Bilateral Filter for 3D Surfaces **28**

4.1 Extension to 3D Surfaces 28

4.2 Predictors 30

4.3 Feature Preservation 32

4.4 Normal Estimation and Improvement 35

4.4.1 Smoothing Triangle Models 35
4.4.2 Improving Normals 37
4.4.3 Normal Transformation by the 3D Bilateral Filter 38
4.4.4 Using Normals for Weighting 39

5 Results **41**
5.1 Triangle Models 41
5.1.1 Comparison to Other Methods 47
5.2 Normal Improvement 53
5.3 Normal Improvement for Mollification 56
5.4 Using Normals for Weighting 58

6 Conclusions **61**

Bibliography **63**

Chapter 1

Introduction

With geometry scanners becoming more widespread and a corresponding growth in the number and complexity of scanned models, *robust* and *efficient* geometry processing becomes increasingly desirable. Even with high-fidelity scanners, the acquired 3D models are invariably noisy [Rusinkiewicz et al. 2002; Levoy et al. 2000], and therefore require smoothing (We use denoising and smoothing interchangeably in this thesis.). Similarly, shapes extracted from volume data (obtained by MRI or CT devices, for instance) often contain significant amounts of noise, be it geometric noise from inaccurate sensors [Taubin 1995; Desbrun et al. 1999], or topological noise due to occlusions or sensor failures [Guskov and Wood 2001; Wood et al. 2002]. This noise must be removed before further processing. Removing noise while preserving the shape is, however, no trivial matter. Sharp features often blurred if no special care is taken in the smoothing process. To make matters worse, meshes from scanners often contain cracks and non-manifold regions. Many smoothing methods diffuse information along the surface, and cannot effectively smooth regions that are not manifold.

In order to state this more formally, we introduce some terminology. Let S be a surface. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ be a set of point samples from the surface. The samples may include other data, such as normals, $N = \{\mathbf{n}_{\mathbf{x}_1}, \mathbf{n}_{\mathbf{x}_2}, ..., \mathbf{n}_{\mathbf{x}_N}\}$, or connectivity information such as a set of neighbors for each sample, or polygons defined on the samples.

The samples X may also be noisy. This noise may be independent from sample to

sample (e.g., due to thermal noise in a CCD), correlated or even systemic (e.g., due to a miscalibration resulting in biased measurements). We will in general assume that the noise is uncorrelated between samples, but not characterized further. Removal of noise that does not match these assumptions is an interesting open problem [Levoy et al. 2000].

Given the samples X, there are several interesting issues. The *surface reconstruction problem* is to estimate S from X. Some reconstruction algorithms operate directly on noisy samples. However, some reconstruction methods (e.g. [Amenta et al. 1998]) do not remove noise during reconstruction; instead, the noise must be removed before or afterwards. Noise can be removed prior to reconstruction by forming a denoised estimate $\bar{\mathbf{x}}_i$ for each sample $\mathbf{x}_i$. Forming such estimates is the goal of this thesis. In general, we do not address the reconstruction problem, though we do discuss how some related work attempts to simultaneously reconstruct S while smoothing out the noise from X.

There are several methods for performing denoising depending on what other information, such as normals or connectivity, is available. Many of these are directly analogous to methods from image processing. However, as in image processing, naive methods are generally not *feature preserving*. In other words, if these methods are used for smoothing, they will not preserve the edges and corners of S. In this thesis, we present smoothing methods that respect features defined by tangent plane discontinuities in S (i.e. edges and corners). This is not the only possible definition of feature: A change in fine-scale texture on S could be perceived as a visual boundary without a large change in surface tangent. This is a much more general problem, which we do not address.

We also note an inherent ambiguity in present in surfaces but not images: images have a natural separation between location and signal, but on surfaces, the location and signal are intertwined. We will discuss this ambiguity further in the following chapter.

1.1 Our Goal

Goal: Given noisy samples X of a surface S, estimate denoised samples $\bar{X}$ while preserving the features of S captured by X.

This problem is difficult since the only information we have about S is contained in the samples X, which are corrupted by noise. Thus, features that create variations smaller than those due to noise are likely to be washed out in the noise. More generally, the problem is inherently ambiguous, since there is no universal definition of a feature, and nothing in the samples to distinguish surface variations due to features versus those caused by noise. Thus, it is necessary to make some assumptions in order to denoise the samples X. In particular, we will assume that variations due to noise occur at a significantly smaller scale than those from features. This assumption is not too restrictive; as will be seen, the only penalty when it is violated is that small features are smoothed away with the noise, while large features will still be preserved. We also assume that S is "reasonably" well sampled by X, though we do not provide a formal definition of "reasonable". In rough terms, we require that the sampling rate is sufficient for an accurate reconstruction of S. These are similar and related to the requirements for sampling and reconstructing signals, such as the Nyquist limit, though not in a formal sense. For a more formal treatment of sampling requirements for surface reconstruction, we refer the reader to [Bernardini and Bajaj 1997; Amenta et al. 1998].

1.2 Approach

We cast feature preserving denoising of 3D surfaces as a robust estimation problem on sample positions. For each sample, we form a robust estimate of the surface based on surrounding samples. Moving a sample to the local surface estimate smooths the samples while preserving features.

There are many possible methods for robust estimation of the type above. We choose to extend the bilateral filter to 3D data. The bilateral filter is a nonlinear,

feature preserving image filter, proposed by Smith and Brady in 1997 [Smith and Brady 1997] and Tomasi and Manduchi in 1998 [Tomasi and Manduchi 1998]. We discuss it and its relation to robust estimation in Chapter 3.

Our discussion thus far has been very general, in order to avoid artificially limiting the applicability of our methods. However, we will demonstrate our methods (and those of the similar approaches mentioned above) on a variety of forms for X: points, points with (noisy) normal information, polygonal meshes, and "polygon soups". As will be seen, our methods are sufficiently general to easily move between these various representations.

Chapter 2

Related Work

There has been a great deal of previous work in the area of isotropic and feature preserving smoothing, both of images and of 3D models. Many model smoothing methods are extensions from the image domain (as is ours). The following discussion is organized around these broad classes of image and surface smoothing techniques.

2.1 Smoothing Without Preserving Features

Noise can be removed from an image by straightforward, isotropic filtering. For example, convolution with a Gaussian filter will remove noise in an image very effectively. It can be shown that the result of an isotropic diffusion process is equivalent to convolution by a Gaussian of a particular width (discussed further in chapter 3.), [Strang 1986], and vice-versa. This connection to diffusion processes (for heat, intensity, or any other sort of information) provides for a straightforward extension to 3D from the image domain.

Fast smoothing of polygonal meshes was initiated by Taubin [1995]. His method considered smoothing as a signal processing problem, and applied filter design techniques from the 1D domain to create an iterative diffusion process over a polygonal mesh that was efficient, simple, and energy-preserving (i.e., non-shrinking). His method could accommodate general linear constraints on the smoothed mesh, including interpolation and tangent constraints, but was not generally feature preserving.

Desbrun et al. [1999] extend Taubin's approach with an implicit solver and a diffusion process over curvature normals. The implicit solver leads to a more stable and efficient solution of the diffusion process. Use of curvature information (rather than vertex positions) prevents vertex drift in areas of the mesh that are sampled in an irregular fashion.

2.2 Anisotropic Diffusion

Smoothing with convolution or, equivalently, isotropic diffusion are not generally feature preserving. This is because features (large variations) and noise (small variations) are treated identically,

Anisotropic diffusion, introduced by Perona and Malik [1990], extends isotropic diffusion with a nonlinear term limiting diffusion across boundaries defined by large gradient in image intensity. The effect is to smooth within regions of the image demarcated by features, but not across them. There has been a significant amount of work analyzing and extending anisotropic diffusion. In particular, we will discuss the relation between anisotropic diffusion and robust statistics, as established by Black et al. [1998], in chapter 3.

The extensions of anisotropic diffusion to meshes are less straightforward and more plentiful than those of isotropic diffusion. In part, this is because feature preserving smoothing is more useful, and so has received a greater amount of attention. Also, there are a number of ways to define features on 3D meshes as compared to images, leading to a wider variety of approaches. Also, as mentioned in the previous chapter, there is an inherent ambiguity in images versus 3D surfaces: in images, signal is well separated from position in the image. On manifolds, the signal and spatial position are deeply intertwined. There are several approaches to resolve this ambiguity, again multiplying the options for extending anisotropic diffusion to 3D.

Desbrun et al. [2000] generalize the mesh-based curvature flow approach [Desbrun et al. 1999] to general bivariate data, including meshes represented as height fields, also extending it with an anisotropic term based on curvature to make it feature pre-

serving. In a more general mesh setting, Clarenz et al. [2000] introduce a nonlinear diffusion tensor at each point on a smooth surface, then form a finite element discretization to solve for the anisotropic curvature diffusion on a polygonal mesh. Bajaj and Xu [2003] further generalize the approach of Clarenz et al. [2000] to 2D manifolds embedded in arbitrary dimension, e.g. meshes with attributes such as color.

Belyaev and Ohtake implement anisotropic diffusion in the normal field of a mesh in an iterative two-step process [2001]. In the first step, normals are updated as nonlinear weighted combinations of normals from adjacent faces. In the second step, vertex positions are updated by projection to the new facets defined by the update normals (see also [Yagou et al. 2003]). Taubin [2001] uses a somewhat similar approach for anisotropic mesh smoothing, but rather than alternating between updating the normal field and vertex positions, he first computes smoothed normals, then uses the smoothed normals to compute weights relating vertex positions, used to create a linear, iterative Laplacian smoothing algorithm.

Anisotropic diffusion directly on meshes continues to generate interest. Meyer et al. [2002] use their newly introduced discrete differential geometry operators to define anisotropic weights to control curvature flow. Working in the domain of level sets, Tasdizen et al. [2002] anisotropically smooth the normal field on the surface, then adapt the surface to match the normal field (similar to Taubin's approach [2001]). There are also several extensions of anisotropic diffusion to 3D volume data and level sets (e.g., [Preusser and Rumpf 2002]), which, though they can be used to smooth 3D models, are only tangentially related to our work.

Anisotropic diffusion is effective for feature preserving denoising of images, and reasonably simple, though some care must be taken in its implementation [You et al. 1996]. It also can be computationally expensive. Its extension to 3D surfaces can be complicated because of the lack of a straightforward mapping between images and surfaces. One shortcoming we address in this thesis is its lack of generality; diffusion processes require a connected domain, which limits its applicability to many surface representations, such as point clouds.

2.3 Wiener Filtering

The Wiener filter from signal processing is designed to simultaneously remove blurring and noise from a signal. Reconstructing a signal from a blurred version of the signal can be achieved by a straightforward deconvolution. However, deconvolution magnifies high frequencies, and thus overenhances any noise degrading the blurred signal. Wiener filtering recosntructs blurred, noisy signals optimally under a least-squares error norm, by trading off the amount of deconvolution versus noise removal. For this reason it is often called *optimal filtering*. Wiener filtering is feature preserving, though a tradeoff between blurring and denoising is present. It is not a fully automatic method, as the spectrum of the noise has to be estimated before filtering.

Extensions to Wiener filtering include locally-adaptive filtering in the wavelet domain, as proposed by Mihçak et al. [1999] and Moulin and Liu et al. [1999], further developed by Strela et al. [2000]. In these methods, features are preserved by considering image behavior at multiple scales. These in turn inspired an extension to the mesh domain [Peng et al. 2001], via a Laplacian-pyramid decomposition based on Loop subdivision [Zorin et al. 1997]. A more direct approach is taken by Alexa [2002], where different smoothing and edge-enhancing effects can be controlled through the choice of an autocorrelation function on the vertex positions. Pauly and Gross [2001] apply Wiener filtering to 3D point models by resampling point positions to a plane as a height field, then treating it as an image.

Wiener filtering is a good alternative to anisotropic diffusion for image and surface denoising. In particular, it is much more efficient. Unfortunately, it shares some of the same limitations as diffusion, such as requiring a connected domain. In the case of surfaces, some extensions even require a semi-regular sampling. Again, these limit its general applicability.

2.4 The Bilateral Filter

The bilateral filter is a nonlinear image filter proposed by Smith and and Brady [1997] and separately by Tomasi and Manduchi [1998]. At each pixel, the denoised estimate is computed as a weighted combination of the surrounding pixels. The weight of a pixel in the estimation falls off as its distance grows (as in the usual non-feature preserving smoothing filters), as well as the difference in intensity from the initial estimate. Thus, pixels separated by large discontinuities in the image are downweighted in each other's estimate. Durand and Dorsey [2002] establish a connection between the bilateral filter and robust statistics, as well as speed up the bilateral filter with linear approximation and other optimizations. Barash [2001; 2002] makes a similar connection to adaptive smoothing, and between adaptive smoothing and the bilateral filter. Elad [2002] explores the theoretical basis of the bilateral filter, finding that it can be considered a single iteration of a well-known minimization method in a Bayesian approach.

There are three extensions of the bilateral filter to 3D surfaces thus far. Our work [Jones et al. 2003] extends the bilateral filter to "polygon soups" by way of first-order predictors based on facets. Concurrently, Fleishman et al. [2003] extend bilateral filtering to meshes, estimating vertex normals via averaging, and using local frames to reduce vertex position estimation to a 2D problem. Choudhury and Tumblin [2003] use a similar approach, but with a more complicated process to estimate the vertex normals. Their method is based on an extension for image smoothing of the bilateral filter with local-frames (see also Elad's [2002] suggestions for improving the bilateral filter). The adaptive smoothing of Ohtake et al. [2002] is similar, but instead smooths the normal field with an adaptive filter adjusted to a local variance estimate, then fits the mesh to the normals (similar to their previous approaches [Belyaev and Ohtake 2001]).

The bilateral filter provides many of the benefits of anisotropic diffusion, but is generally more efficient. As will be discussed in the next chapter, it does not rely on a connected domain, and also provides for a reasonably simple extension to 3D

meshes. Since the bilateral filter forms the basis for our methods, we discuss it and its extensions to 3D surfaces in more detail in chapters 3 and 4, respectively.

2.5 Surface Reconstruction Methods

Some methods for reconstructing surfaces from point clouds or other scanned data include a component of noise removal, either implicit or explicit. Soucy and Laurendeau [1992] compute confidence-weighted averages of points, then triangulate the smoothed points. The algorithm of Turk and Levoy [1994] aligns and zippers multiple range images, then resamples vertices to a confidence weighted average from the range images. The Vrip algorithm [Curless and Levoy 1996] converts range scans into distance volumes, that are weighted by a confidence metric and merged. They then extract an isosurface from the merged volume. Wheeler et al. [1998] use a similar approach with refinements to eliminate some biases in the Vrip algorithm, and to better handle outliers in the data. Although all of these methods remove noise by averaging together multiple estimates of the surface, they do not preserve features.

Radial basis functions can be used to fit point data [Carr et al. 2001] with error tolerances, which can be used for noise removal if the magnitude of measurement error can be estimated.

Moving Least Squares [Levin 2001; Alexa et al. 2001; Alexa et al. 2003] computes surfaces from point clouds as the fixed-points of a projection operator. The use of a smooth weighting function removes noise, and though the method is not explicitly feature preserving, it is a local robust estimator (cf. chapter 3).

Finally, polygonal meshes can be smoothed if they are watertight and manifold, by scanning them into scalar volumes, e.g. via a distance transform, and then applying a 3D smoothing filter, and re-extracting the surface from the volume [Šrámek and Kaufman 1999].

2.6 Discussion

As can be seen, most image and surface smoothing methods that operate directly on the image or surface are limited to connected domains. Some, such as MLS, operate on independent samples, but are not feature preserving. An exception is the bilateral filter, which does not rely on adjacency information in the image. This independence from connectivity will prove advantageous in the following chapters. We discuss the bilateral filter in more depth in the following chapter, and its extensions to 3D surfaces in chapter 4.

Chapter 3

Robust Estimation, Anisotropic Diffusion and the Bilateral Filter

In order to provide a base for our approach to feature preserving smoothing, it is necessary to discuss robust estimation, along with its connection to anisotropic diffusion and the bilateral filter for images. Most of this discussion is a combination of the work by Black et al. [1998], Durand and Dorsey [2002], Elad [2002], and Barash [2001].

3.1 Robust Estimation

Robust estimation is concerned with forming estimates from data in the presence of *outliers*, i.e. data that violate the assumptions of our theoretical model. For example, if we encounter the data in figure 3-1, a reasonable explanation is that the process generating the data is linear with some source of Gaussian noise. If we consider the data in figure 3-2, it is still reasonable to assume that the generating process is linear, despite the presence of values that are obviously not due to a linear process (i.e. outliers), perhaps due to operator or transcription error.

Standard techniques for fitting model parameters will succeed on clean data such as in figure 3-1, since none of the points deviates far from the theoretical model of the data. However, they fail when outliers are present. If we use a least-squares

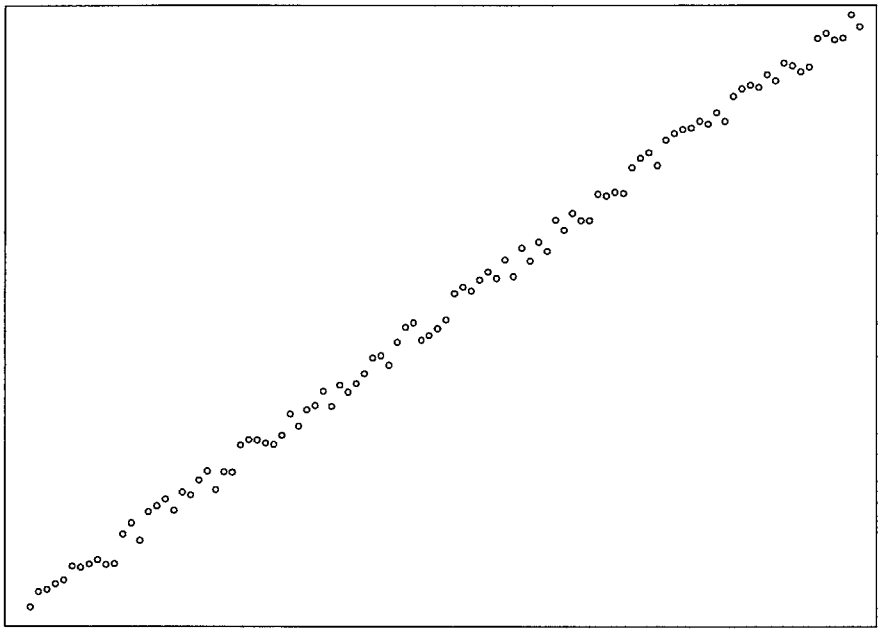


Figure 3-1: Data from a linear process with noise.

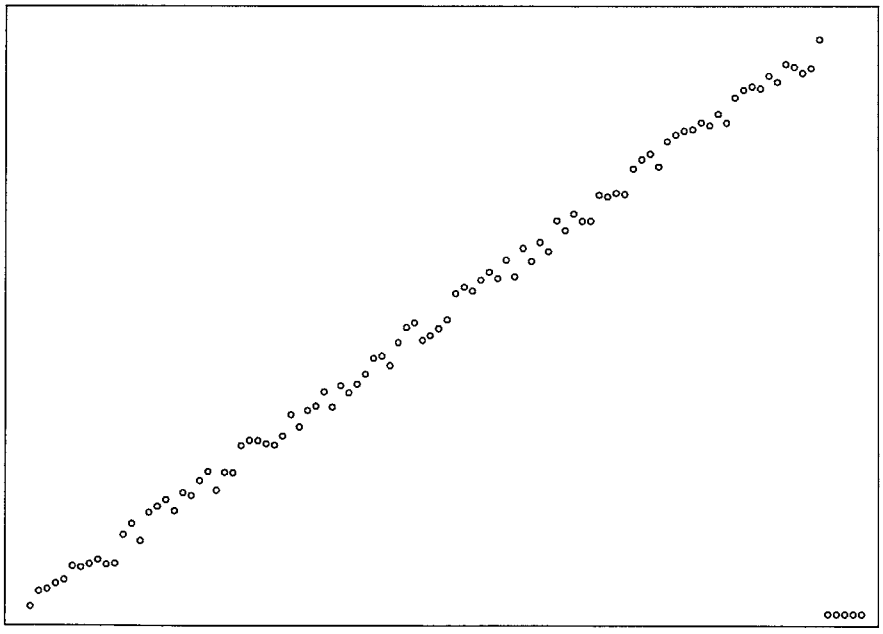


Figure 3-2: Data from a linear process with noise, and a few outliers in the bottom right.

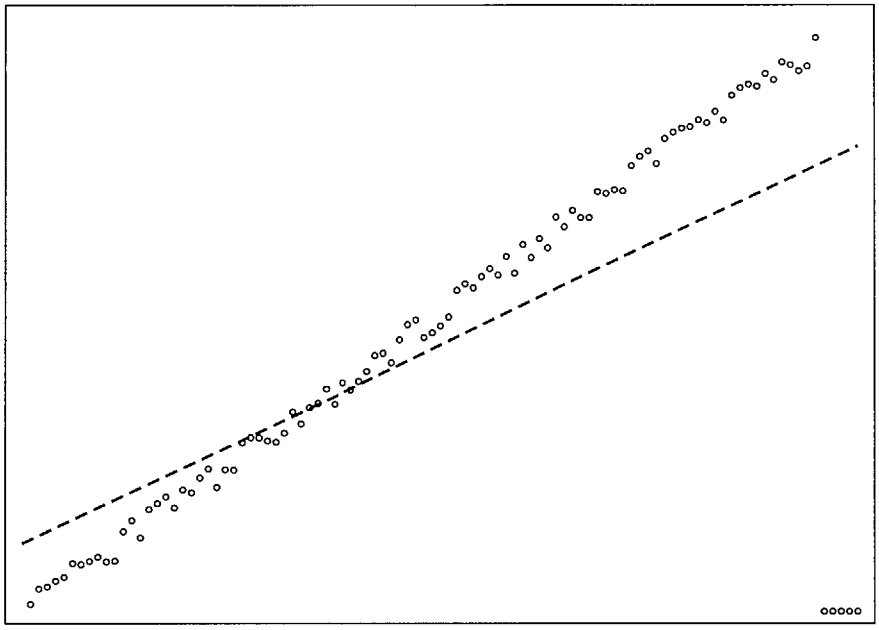


Figure 3-3: Least Squares fit to data from figure 3-2.

error minimization to fit a line to the data in figure 3-2, the result is the poor fit shown in figure 3-3. In fact, a single outlier can force the least-squares estimation to an arbitrary position, depending on its value. If we examine the equation for least-squares fit, the reason for this failure becomes obvious.

The fitting of parameters $\bar{\mathbf{y}}$ for a model to the data points X can be expressed as a minimization problem,

$$\bar{\mathbf{y}} = \arg\min_{\mathbf{y}} \sum_{i=1}^{N} |d(\mathbf{y}, \mathbf{x}_i)|^2, \tag{3.1}$$

with $d(\mathbf{y}, \mathbf{x})$ the distance between the actual value and the prediction for $\mathbf{x}$ in a model with parameters $\mathbf{y}$. As the distance between the prediction and the measured datum increases, the error grows without bound, as in figure 3-4. Thus, a single outlier can have as much influence over the minimization as the rest of the data.

In order to address this shortcoming, it is necessary to use some other approach. One way to make (3.1) robust is to use a robust error norm [Hampel et al. 1986],

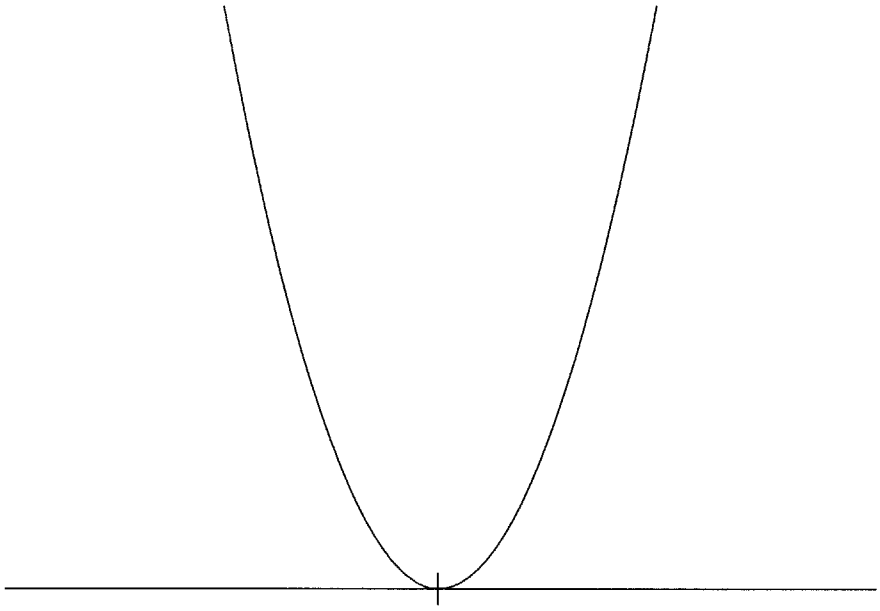


Figure 3-4: Least-squares error norm: d^2.

$\rho(d)$, as in

$$\bar{\mathbf{y}} = \arg\min_{\mathbf{y}} \sum_{i=1}^{N} \rho(d(\mathbf{y}, \mathbf{x}_i), \sigma), \tag{3.2}$$

In order for this estimator to be robust, the derivative of the norm, $\rho'(d)$, should be bounded above and below. This prevents a single datum from influencing the estimate to an arbitrary extent . It should also be zero when a prediction matches its datum, i.e. $d(0) = 0$, and be monotonic increasing with $|d|$, so that worse estimates have larger (or at least not less) error under the norm. An example of a function meeting these criteria is a Gaussian error norm,

$$\rho(d) = 1 - exp(-\frac{d^2}{\sigma^2}), \tag{3.3}$$

shown in figure 3-5. The scale parameter σ controls the width of the central "valley" in the norm, and therefore how quickly data are treated as outliers, limiting their influence over the error minimization. If we perform the same fit to the data in figure 3-2, but using the robust norm from (3.3), the result is figure 3-6. The fit is much more reasonable, as a small set of outliers cannot affect the estimate arbitrarily.

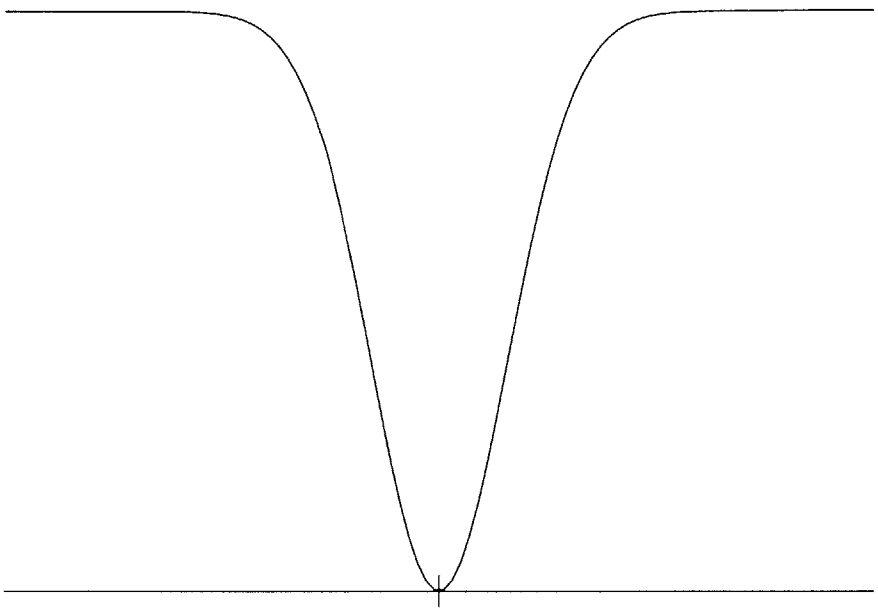


Figure 3-5: Robust (Gaussian) error norm: $\rho(d) = 1 - exp(-\frac{d^2}{\sigma^2})$.

The Gaussian error norm is of course not the only possible choice for a robust norm. There are several other functions that meet the criteria above, but we have found the Gaussian norm sufficient for our experiments. We refer the interested reader to more in-depth discussions [Huber 1981; Hampel et al. 1986; Black et al. 1998].

3.2 Anisotropic Diffusion

Diffusion processes as used in image processing are methods of smoothing data by local operations on small groups of pixels. "Diffusion" in images refers by analogy to diffusion of heat in a material. For the case of a uniform material heat diffusion is governed by the heat diffusion equation,

$$\frac{\partial T}{\partial t} \propto -\nabla \cdot (\nabla T). \tag{3.4}$$

In simple terms, the change in temperature is proportional to the divergence of the temperature gradient. The diffusion equation can be discretized in space and time to

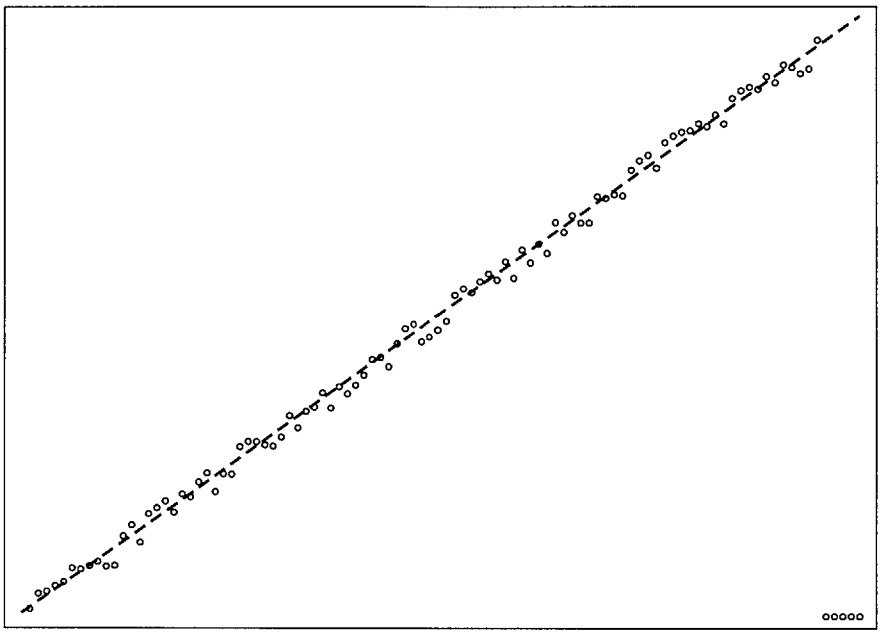


Figure 3-6: Robust fit to data from figure 3-2.

give the change in temperature per time step,

$$I_s^{t+1} = I_s^t + \lambda \sum_{p \in \eta_4(s)} I_p^t - I_s^t, \tag{3.5}$$

where s and p are pixel positions, $\eta_4(s)$ is the 4-neighborhood of the pixel at s, λ is a scalar controlling the rate of diffusion, and t is a discrete time step. The isotropic diffusion equation for images is formed by direct analogy to the heat diffusion equation (3.5), with intensity replacing heat.

Applying several iterations of the discrete diffusion equation to an input image results in a smoothed but blurred version of that image, as shown in figures 3-7 and 3-8. Those readers familiar with image processing will probably find the appearance of the result of diffusion familiar, as it approximates a Gaussian blur operation. This is not coincidence; the solution of the heat diffusion equation can be found by convolution with a Gaussian function [Fourier 1955].

It is obvious that isotropic diffusion is not feature preserving. A large change in intensity between adjacent pixels is treated the same as a small variation, so features

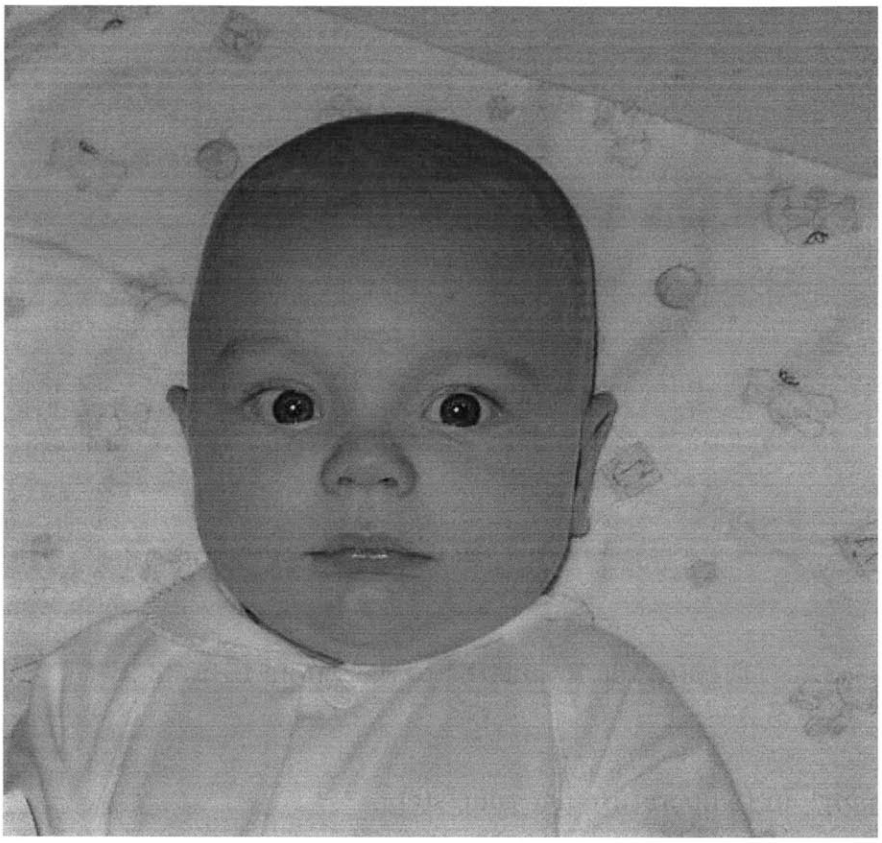


Figure 3-7: Input image for isotropic and anisotropic diffusion.

of the image and noise will be smoothed out identically. To address this shortcoming, Perona and Malik [1990] proposed modifying the isotropic diffusion equation to make it feature preserving,

$$I_s^{t+1} = I_s^t + \lambda \sum_{p \in \eta_4(s)} g(I_p^t - I_s^t)\,(I_p^t - I_s^t), \tag{3.6}$$

where $g(x)$ is an *edge-stopping function* that inhibits diffusion across strong edges in the image. They propose two such functions,

$$g_1(x) = \frac{1}{1 + \frac{x^2}{\sigma^2}} \quad \text{and} \quad g_2(x) = e^{-(x^2/\sigma^2)}, \tag{3.7}$$

where σ is a parameter controlling how large a change in pixel intensity is considered

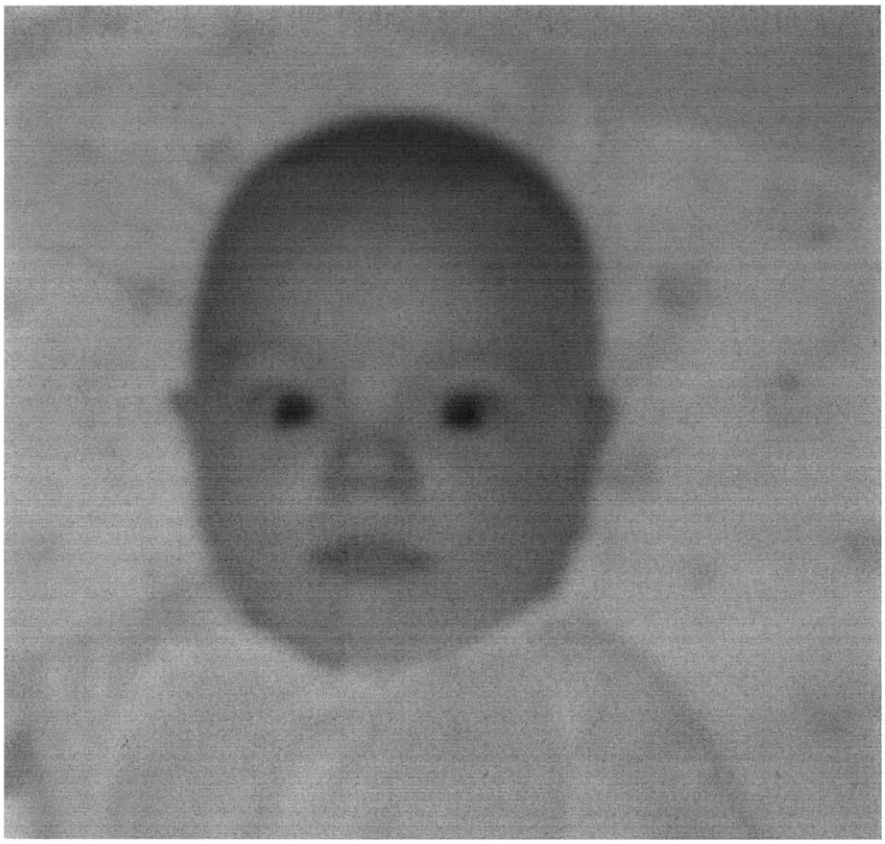


Figure 3-8: Isotropic diffusion applied to figure 3-7.

an edge, and as such should limit diffusion.

Applying anisotropic diffusion with the edge-stopping function $g_2(x)$ to our previous example yields a much better result, in terms of feature preservation, as shown in figure 3-9. Areas with small variations have been smoothed out, removing the noise and minor textural variations. Sharp edges have been preserved, for example around the pupils and edge of the face.

3.3 Anisotropic Diffusion and Robust Estimation

Black et al. examine anisotropic diffusion in terms of robust estimation. Their key insight is that anisotropic diffusion can be seen as the solution to a robust estimation problem at each pixel, and that in particular, "boundaries between ... regions are

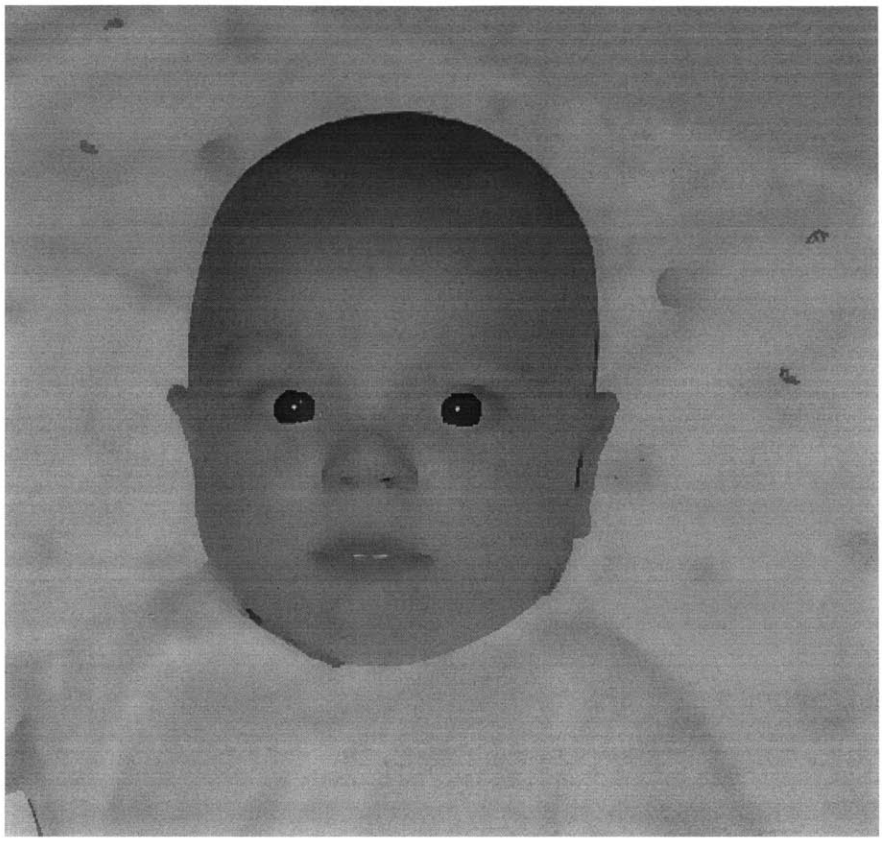


Figure 3-9: Anisotropic diffusion applied to figure 3-7.

considered to be outliers" [Black et al. 1998]. Much of this section follows their discussion.

Consider estimating a piecewise constant image from noisy pixel data. This can be expressed as an optimization problem as

$$\min_I \sum_s \sum_{p \in \eta_4(s)} \rho(I_p - I_s, \sigma), \tag{3.8}$$

where $\rho(\cdot)$ is a robust error norm with scale parameter σ. An I minimizing (3.8) will have adjacent pixels with similar intensity values. The effect of edges, where $(I_p - I_s)$ is large, on the minimization will be limited by the use of a suitable robust error norm. The use of a robust norm causes all edges with a larger change than some threshold controlled by σ to be treated equivalently in the minimization.

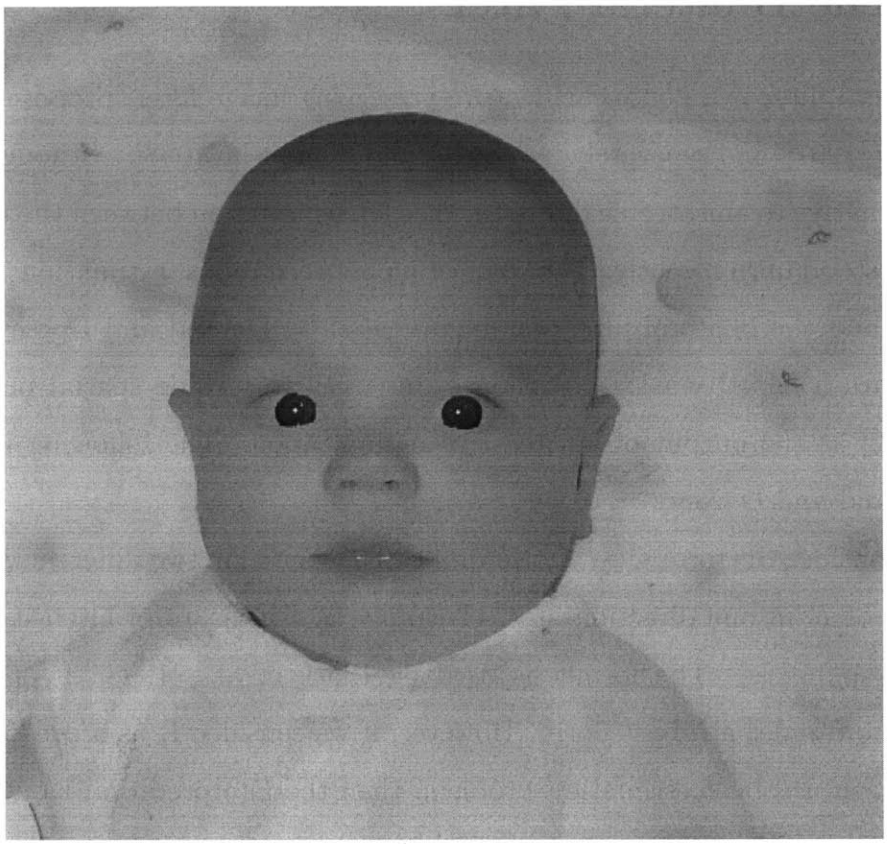


Figure 3-10: The bilateral filter applied to figure 3-7.

Using gradient descent to solve (3.8) gives

$$I_s^{t+1} = I_s^t + \lambda \sum_{p \in \eta_4(s)} \psi(I_p^t - I_s^t, \sigma), \tag{3.9}$$

where $\psi(\cdot) = \rho'(\cdot)$. If the initial estimate for the gradient descent is our noisy image, then we can see a strong similarity between discrete anisotropic diffusion and gradient descent, made explicit if we define the edge-stopping function in (3.5) in terms of the robust error norm in (3.9), i.e. $g(x) \equiv \rho'(x)/x$. This fundamental connection between the two methods allows much of the previous work in anisotropic diffusion and robust estimation to be considered as two expressions of the same methodology.

3.4 The Bilateral Filter

The bilateral filter is a nonlinear, feature preserving image filter, proposed by Smith and Brady [1997], and separately by Tomasi and Manduchi [1998]. Although proposed as an alternative to anisotropic diffusion, the close connection between the two was not well understood until recently. The connection between robust estimation, anisotropic diffusion, and the bilateral filter was investigated by Durand and Dorsey [2002], as well as Barash [2001] via an extension of intensity to include spatial position, and Elad [2002] as the output of an iterative minimization. The following is primarily from Durand and Dorsey.

If we consider the term $I_p^t - I_s^t$ in (3.6), it can be viewed in two different ways: as the derivative of I^t in one direction at s, or simply as a measure of intensity difference between two pixels. The former is used by Black et al. [1998], in a variational minimization under a robust norm. However, if we consider I_p as a predictor for the value of I_s in a robust estimation problem, then the interpretation as a measure of difference is more intuitive.

If we consider (3.6) as solving a robust estimation problem with local information, it is a small step to consider extending the neighborhood of the estimation and using a spatial weighting function $f(\cdot)$,

$$I_s^{t+1} = I_s^t + \lambda \sum_{p \in \Omega} f(p - s)\, g(I_p^t - I_s^t)\, (I_p^t - I_s^t), \tag{3.10}$$

where Ω is the domain of the image, and $f(\cdot)$ is a spatial weighting function (which in (3.6) would be unity for a pixel's 4-neighborhood and zero elsewhere).

If we solve for the fixed point I' of (3.10) by setting the gradient step to zero, we get the bilateral filter[1],

$$I'_s = \frac{1}{k(s)} \sum_{p \in \Omega} f(p - s)\, g(I_p - I_s)\, I_p, \tag{3.11}$$

[1]This derivation is from Durand and Dorsey [2002], section 4.1.

$f(\cdot)$ is the spatial falloff function, and $g(\cdot)$ the falloff function in the intensity domain. Usually, Gaussians are used for $f(\cdot)$ and $g(\cdot)$. The normalization term $k(s)$ is the sum of the product of the spatial and intensity weighting functions,

$$k(s) = \sum_{p \in \Omega} f(p - s)\ g(I_p - I_s). \tag{3.12}$$

Examining the bilateral filter, we see that it appears similar to filtering by convolution, in that output pixels are weighted average of the input. However, in addition to weighting pixels according to distance, as in convolution, it includes a term that takes into account the difference in intensities. By the derivation above, functions that perform well as edge-stopping functions should also do so as intensity weight functions in the bilateral filter. The connection between bilateral filtering, anisotropic diffusion, and robust estimation allows one to use concepts and ideas from any one area in the other two. The effect of the bilateral filter on an image is demonstrated in figure 3-10.

It has been noted empirically that bilateral filtering generally becomes close to a fixed point in a single iteration [Tomasi and Manduchi 1998; Durand and Dorsey 2002; Choudhury and Tumblin 2003]. Thus, it belongs to the class of *one-step W-estimators* or *w-estimators* [Hampel et al. 1986]. This leads us to believe that a single application of the bilateral filter should be sufficient for smoothing, whether an image or a 3D surface. This belief will be borne out by experiments.

In order to make our extension to 3D conceptually simpler, we will modify (3.11), slightly, by introducing the concept of a *predictor*. In the image domain, given the information at pixels s and p, the prediction for s due to p will be written as $\Pi_p(s)$. Incorporating this into the bilateral filter gives

$$I'_s = \frac{1}{k(s)} \sum_{p \in \Omega} f(p - s)\ g(\Pi_p(s) - I_s)\ \Pi_p(s). \tag{3.13}$$

In its original incarnation, the bilateral filter uses a zeroth-order predictor, $\Pi_p(s) = I_p$. In the next chapter, we will show how using more powerful predictors allows us to extend bilateral filtering to 3D data.

Chapter 4

Bilateral Filter for 3D Surfaces

We return to our problem statement from section 1.1.

Goal: Given samples X, corrupted by noise, of a surface S, estimate denoised samples $\bar{X}$ while preserving the features of S captured by X.

Our samples could be point samples scattered on the surface, or points with normals, or small triangles or patches. We assume that all information in the samples is noisy, including position, normal, and connectivity (if present). Our goal is to remove the noise from the position and normal information; we do not attempt to fix topological errors.

4.1 Extension to 3D Surfaces

The success of the bilateral filter for image denoising encourages us to use similar methods for surfaces. This follows the pattern of much of the previous work in which image techniques have been successfully extended to polygonal meshes and other 3D data (as discussed in chapter 2).

Such extensions are nontrivial because of a fundamental difference between 2D images and 3D surfaces: Images have an inherent parameterization that is well separated from the image's signal. In the usual case, pixels are arranged on a rectangular grid, giving a simple and global parameterization. This separation of position vs. signal (or domain vs. range) in images simplifies operations such as smoothing and

feature detection. Surfaces in 3D lack this conceptual separation. The spatial position of a point on a surface and the surface's "signal" at that point are the same. We must find some way to separate position and signal to apply methods from image processing.

This separation can be achieved with the conceptual adjustment to the bilateral filter presented at the end of the previous chapter. Rather than filtering based on sample positions alone, we will extend the filter to use first-order predictors by taking advantage of normal information. This is similar to one of the most common methods of extending image processing algorithms to surfaces, that of using normal information at each point on the surface to form a local frame for computations centered on that vertex. The surrounding surface is then treated as an image or height field in the local frame with the central vertex at the origin. Our method generalizes the standard approach, encompassing a wider class of extensions, including the height field approximation.

With normal information available , we are able to define the bilateral filter for 3D,

$$\bar{\mathbf{s}} = \frac{1}{k(\mathbf{s})} \sum_{\mathbf{p} \in X} f(|\mathbf{p} - \mathbf{s}|)\, g(|\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{s}|)\, \Pi_{\mathbf{p}}(\mathbf{s}), \tag{4.1}$$

where $\mathbf{s}$ is a point for which we want to estimate the denoised value $\bar{\mathbf{s}}$, $\mathbf{p}$ is a point which can be used to form a prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ for the denoised position of $\mathbf{s}$. The spatial weight function $f(\cdot)$ controls how wide a neighborhood of samples are used in the estimate $\bar{\mathbf{s}}$, while the influence weight function $g(\cdot)$ controls which predictions are treated as outliers. If a sample $\mathbf{p}$ is far away from s, then its weight will be diminished by the falloff in $f(\cdot)$. Similarly, if the prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ is far from $\mathbf{s}$, it will be receive less weight because of the falloff in $g(\cdot)$. Thus, the estimation for $\bar{\mathbf{s}}$ depends for the most part on samples near $\mathbf{s}$ that also predict locations near $\mathbf{s}$. As will be discussed below, this leads to feature preserving behavior when the filter is applied to 3D data, as predictions from points on opposite sides of features (i.e., across edges and corners) are widely separated.

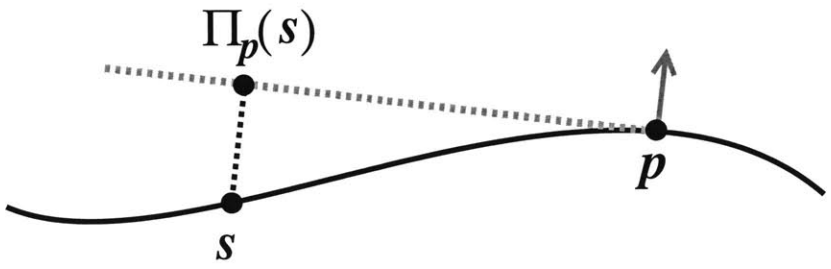
We note that several such extensions have been simultaneously proposed with this

work. This thesis is an extended treatment of our work on filtering triangle models [Jones et al. 2003], in which the bilateral filter is extended to polygonal meshes without connectivity. Fleishman et al. [2003] extend the bilateral filter to (mostly) connected meshes, using an iterative approach. Choudhury and Tumblin [2003] extend the bilateral filter for images to use a type of first-order predictor, then generalize this extension to meshes. We will discuss each of these implementations below.

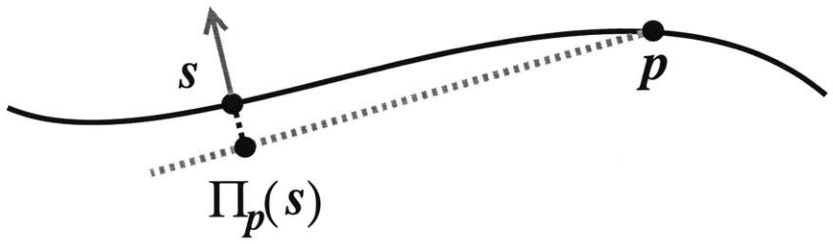
In order to use first order information, such as normals, such information must be present in the input or inferred from it. We will address the two most common representations: polygons and point clouds. In the case of polygons, if the connectivity information is not very noisy, i.e. it matches the topology of the surface closely, then vertex normals can be estimated by weighted averaging of adjacent facet normals; this is the usual approach [Fleishman et al. 2003; Choudhury and Tumblin 2003]. If there is insufficient connectivity, then the facet normals can be used directly, as in our work applied to triangle models [Jones et al. 2003]. If the samples are simply a point cloud without further information, then we must use some method of estimating the normals at the points, such as those used in moving least-squares [Alexa et al. 2001]. Whether normals are already present, inferred from connectivity information, or calculated from point locations, they will probably be noisy or corrupted. For this reason, we also develop methods of improving estimated or preexisting normals, through a novel reuse of the machinery of the filter (section 4.4). For now, we will assume that normals are available, either with each point, or in the case of polygonal samples, for each polygon.

4.2 Predictors

In order to make our extension concrete, we much choose a predictor $\Pi_{\mathbf{p}}(\mathbf{s})$. If we assume that we have points with normals, then there are two natural choices for the predictor, projecting $\mathbf{s}$ to the plane of $\mathbf{p}$, or building a local frame with $\mathbf{s}$ and its normal and predicting $\mathbf{s}$ to have the same height in that local frame as $\mathbf{p}$. The two



(a) Our predictor from Jones et al. [2003].



(b) Predictor from Fleishman et al. [2003]

Figure 4-1: 3D predictors for point **s** due to point **p**.

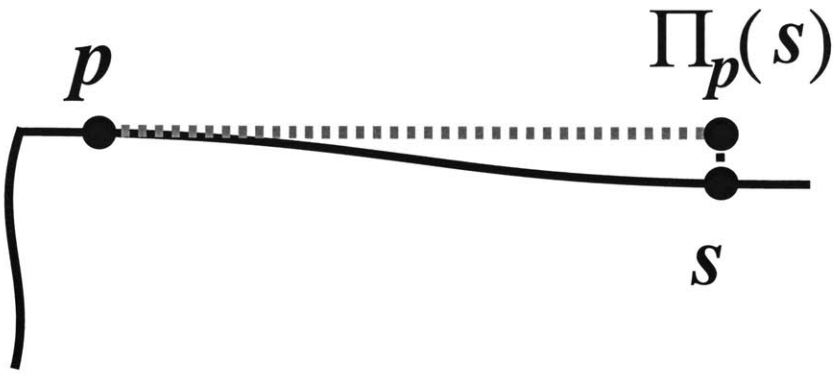


Figure 4-2: When $\mathbf{s}$ and $\mathbf{p}$ are on the same side of the corner, $\Pi_{\mathbf{p}}(\mathbf{s})$ is near $\mathbf{s}$.

predictors can be written

$$\Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + ((\mathbf{p} - \mathbf{s}) \cdot \mathbf{n}_{\mathbf{p}})\mathbf{n}_{\mathbf{p}} \quad \text{and} \quad \Pi_{\mathbf{p}}(\mathbf{s}) = \mathbf{s} + ((\mathbf{p} - \mathbf{s}) \cdot \mathbf{n}_{\mathbf{s}})\mathbf{n}_{\mathbf{s}}, \tag{4.2}$$

respectively. They are illustrated in 4-1. The main difference between the two predictors is which normals are used to form predictions. In the first, each prediction for a point uses a different normal, while the second uses a single normal, the one at the point for which predictions are being made.

More complicated predictors could be conceived, but we will deal primarily with these two. The first is from our previous work [Jones et al. 2003], while the second is used by Fleishman et al. [2003] and Choudhury and Tumblin [2003].

4.3 Feature Preservation

The fact that bilateral filtering is feature preserving in the image domain does not, *per se*, make it feature preserving when extended to 3D. We explore the 3D filter's behavior near corners using the predictors above, with an eye to understanding its treatment of features.

We examine a single corner, separating two mostly flat regions. For now, we assume the surface is noise free. If we consider two samples on the surface, $\mathbf{s}$ and $\mathbf{p}$,

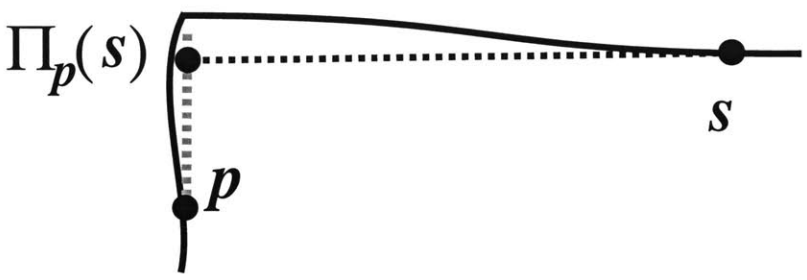


Figure 4-3: When $\mathbf{s}$ and $\mathbf{p}$ are on opposite sides of the corner and Jones et al.'s [2003] predictor is used, the distance between $\Pi_{\mathbf{p}}(\mathbf{s})$ and $\mathbf{s}$ depends mostly on $\mathbf{s}$'s distance from the corner.

both on the same side of the feature, then the prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ is likely to be very close to $\mathbf{s}$, as shown in figure 4-2 (regardless of which predictor we use). The term for the influence weight in the filter for this prediction, $g(\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{s})$, will therefore be near its maximum. The weight of this prediction will therefore depend for the most part on the spatial weight function, $f(|\mathbf{p} - \mathbf{s}|)$.

If we instead consider the case where the two points are across a corner from one another, we see quite different behavior. In this case, the prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ is farther from $\mathbf{s}$. In the case of the first predictor above, the distance between $\mathbf{s}$ and $\Pi_{\mathbf{p}}(\mathbf{s})$ grows proportionally to $\mathbf{s}$'s distance from the corner, as shown in 4-3. If we use the second predictor, it grows as $\mathbf{p}$'s distance from the corner, as shown in 4-4. In both cases, the large separation between $\mathbf{s}$ and the prediction from $\mathbf{p}$ results in the influence weight term, $g(|\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{s}|)$, dominating the filter by downweighting the prediction.

Note that it is always the case that $|\mathbf{p} - \mathbf{s}| \geq |\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{s}|$. Thus, the main difference between the two cases discussed above are how much the influence weight term affects the estimate. When $\mathbf{s}$ and $\mathbf{p}$ are in the same mostly planar region, the bilateral filter behaves similarly to a standard convolution filter, removing noise. When $\mathbf{s}$ and $\mathbf{p}$ are separated by a feature, however, the filter cuts off more quickly, due to the influence weight function. This prevents estimates from different regions of

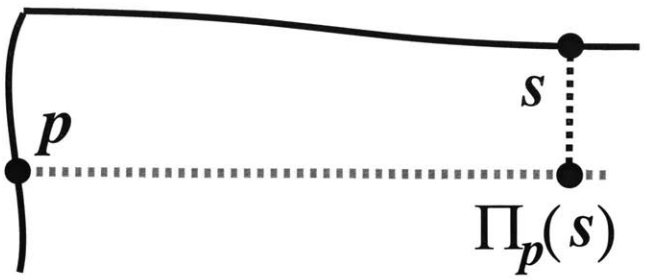


Figure 4-4: When $\mathbf{s}$ and $\mathbf{p}$ are on opposite sides of the corner and Fleishman et al.'s [2003] predictor is used, the distance between $\Pi_{\mathbf{p}}(\mathbf{s})$ and $\mathbf{s}$ depends mostly on $\mathbf{p}$'s distance from the corner.

the surface from "contaminating" the estimate for $\mathbf{s}$. This is what gives the bilateral filter its feature preserving behavior. Another difference between the two predictors is that the predictor or Fleishman et al. does not introduce tangential drift, as all predictions are normal to $\mathbf{s}$. However, it does tend to move vertices in the normal direction to round off corners. If either filter is applied too many times, the estimated shape or the sample positions on it will be degraded. In general, few iterations are required and drift is not a concern.

As a final demonstration of this nonlinear, feature respecting weighting, we show how the spatial $f(\cdot)$, influence weight $g(\cdot)$, and their combination $f(\cdot)g(\cdot)$ behave as $\mathbf{p}$ moves along a curve with a feature, as shown in figure 4-5, while $\mathbf{s}$ remains fixed (at $t = 0.4$ in this example).

The plots for $f(\cdot)$, $g(\cdot)$, and their product are shown in figure 4-6. The sharp corner occurs at $t = 0.5$. As can be seen, while $\mathbf{p}$ is on the same side of the corner as $\mathbf{s}$, i.e. on the first half of the curve ($t_{\mathbf{p}} < 0.5$), the prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ stays close to $\mathbf{s}$. Thus, in the plots we see that $g(\cdot) \approx 1$, and the behavior of $f(\cdot)g(\cdot)$ is governed by $f(\cdot)$.

When $\mathbf{p}$ is on the second half of the curve ($t_{\mathbf{p}} > 0.5$), and therefore on the opposite side of the corner from $\mathbf{s}$, the prediction $\Pi_{\mathbf{p}}(\mathbf{s})$ tends to be much farther from $\mathbf{s}$. Thus,

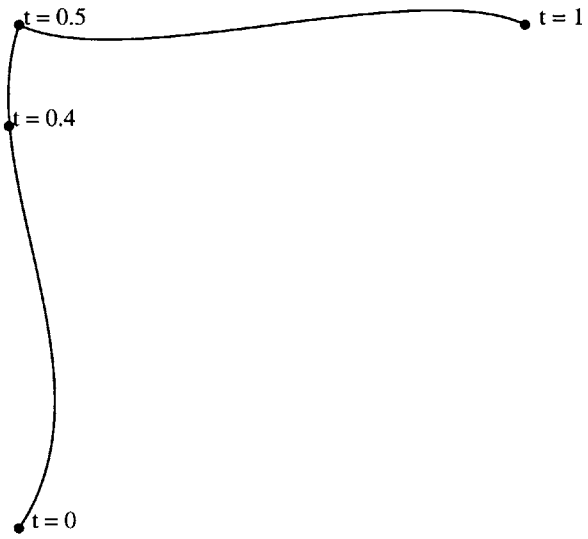


Figure 4-5: A curve with a feature at $t = 0.5$. The plots in figure 4-6 are for $\mathbf{s}$ at $t = 0.4$.

very quickly $g(\cdot)$ drops to near zero, and the product $f(\cdot)g(\cdot)$ is dominated by $g(\cdot)$.

4.4 Normal Estimation and Improvement

The predictors we have discussed require normals (actually, tangent planes) to form predictions. Normals might be provided, or need to be estimated from the data. In either case, they are likely to be noisy, and need some sort of smoothing, or *mollification* [Huber 1981; Murio 1993], before use in predictors. The need for mollification of noisy normals will be demonstrated and discussed further in the next chapter.

4.4.1 Smoothing Triangle Models

In the context of smoothing triangle models without connectivity, we proposed using a predictor based on facets, but applied to vertex positions [Jones et al. 2003], by taking $\mathbf{p}$ as the centroid of the facet. In the limit, this is the same as the approach discussed above for points. As the facet normals are first-order entities of the vertices, noise in the vertices is magnified in the normals and predictions. In order to address this, we applied a mollification pass, using a standard blurring filter on the vertex positions to

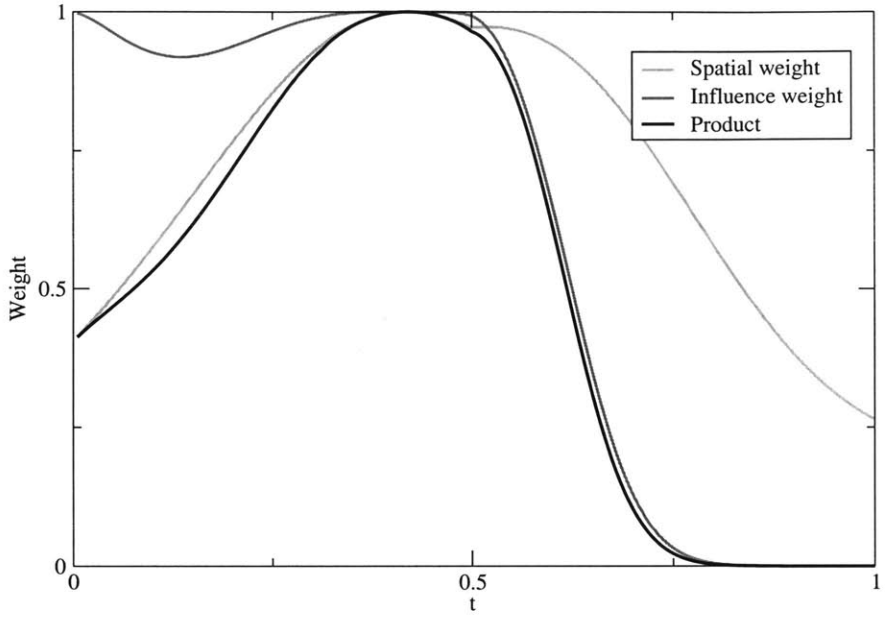
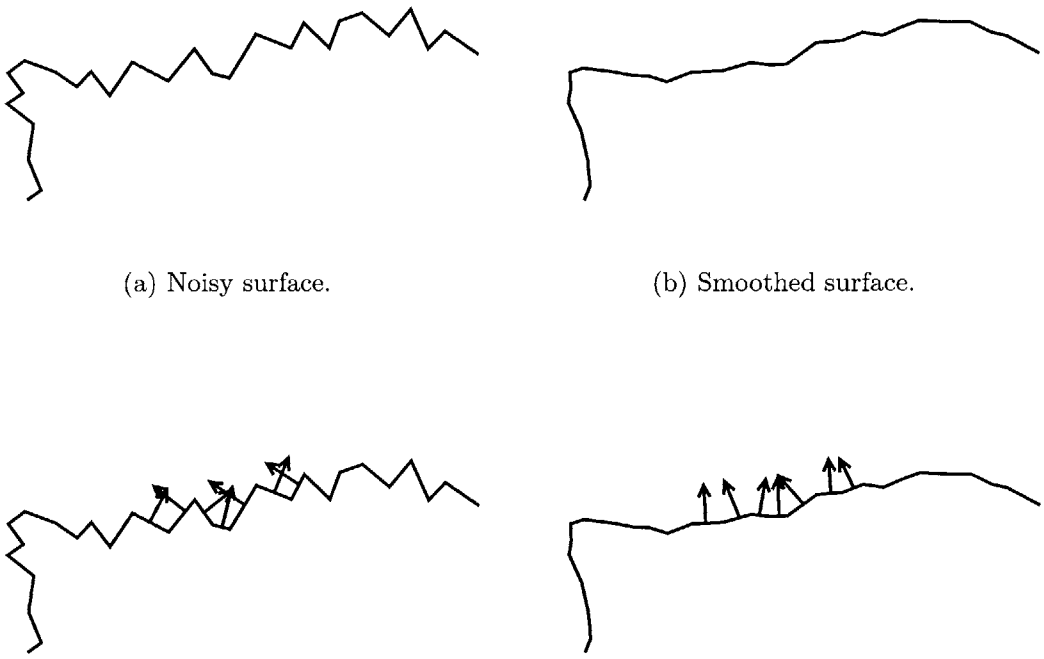


Figure 4-6: Plots for the values of the spatial weight function $f(\cdot)$, influence weight function $g(\cdot)$, and their product $f(\cdot)g(\cdot)$ as $\mathbf{p}$ moves along the curve in figure 4-5 The corner

create a smoothed model (without feature preserving). The original vertex positions were not modified in this pass; instead, the normals of the facets from the smoothed model were copied to the corresponding facets of the original, and these smoothed normals were used for prediction. This method was found to be sufficient for almost all models.

Fleishman et al. take a different approach [2003] in a similar context. They assume a connected triangle mesh, and use area-weighted normals at each vertex. Rather than smooth the normals, however, they perform multiple iterations of the filter to smooth a model. The averaging smooths the normals somewhat, and although the initial predictions are likely to be poor due to noise, as each pass removes some of the noise, the normals and estimations improve.

Choudhury and Tumblin [2003] apply multiple passes to a connected mesh to estimate the normals. They use a modified bilateral filter on the normal data. In a final pass, they estimate vertex positions from the now-denoised normals. Their method is very effective, significantly outperforming our method [2003] on some models with



(a) Noisy surface.

(b) Smoothed surface.

(c) Noisy normals from (a)

(d) Smoothed normals from (b)

Figure 4-7: Smoothing a surface also smooths the normals of the surface.

a high level of noise.

4.4.2 Improving Normals

We propose a new method for improving normals, whether estimated or provided, before applying the 3D bilateral filter to samples of a surface. This new method is based on the 3D bilateral filter, and its denoising properties.

Consider a noisy surface, such as the one shown in figure 4-7(a). If we apply a feature preserving smoothing filter to this surface, the noise will be attenuated, as in figure 4-7(b). Our normal improvement method is based on how this smoothing operation affects the normals of the surface. As shown in figures 4-7(c) and 4-7(d), smoothing a surface also smooths the normals in the sense of making them point in

the same direction in flat areas of the surface. We would like to use these smoothed normals in the predictor function. We do not necessarily have an actual surface to smooth and then extract normals from. However, we can smooth the normals as if they were part of a continuous surface to which we were applying the 3D bilateral filter, via the adjoint of the 3D bilateral filter.

4.4.3 Normal Transformation by the 3D Bilateral Filter

We reconsider the 3D bilateral filter 4.1 as a spatially varying deformation $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. In this interpretation, the denoised estimate for a sample $\mathbf{s}$ is

$$\bar{\mathbf{s}} = F(\mathbf{s}), \tag{4.3}$$

with the summation and other terms incorporated into F.

As discussed by Barr [1984], "a locally specified deformation modifies the tangent space [of an object subject to the deformation]." We can determine how the 3D bilateral filter affects a normal $\mathbf{n_s}$ located in space at $\mathbf{s}$ via the Jacobian J of F at $\mathbf{s}$, computed as

$$J_i(\mathbf{s}) = \frac{\partial F(\mathbf{s})}{\partial s_i}, \tag{4.4}$$

where J_i is the i^{th} column of J, and s_i is the i^{th} spatial component of $\mathbf{s}$.

The normal transformation rule is given by Barr [1984, eq. 1.3] as

$$\bar{\mathbf{n}}_\mathbf{s} = |J(\mathbf{s})| J(\mathbf{s})^{-T} \mathbf{n_s}, \tag{4.5}$$

where $|J|$ is the determinant of J, and $J(\mathbf{s})^{-T}$ is the inverse transpose of J. Since only the direction of the normal is important, $|J|$ can be dropped from the equation and the adjoint of J used instead of the inverse, if $\bar{\mathbf{n}}_\mathbf{s}$ is renormalized afterwards.

Our normal improvement algorithm is simply to compute $J(\mathbf{s})^{-T}$ at each sample $\mathbf{s}$ and apply (4.5) to the corresponding estimated normals, either until the normals converge or for some fixed number of iterations. After the normals have been denoised, a single application of (4.1) is used to estimate vertex positions using. It is important

to note that $J(\mathbf{s})$ is recomputed with the normals from the previous iteration for each step of normal improvement, and that vertex positions are not modified by the normal improvement steps, and also that this normal improvement obviates the mollification pass. We are unable to prove the convergence of this algorithm at this time, but explore its behavior in the following chapter.

There are limitations to this method. For smoothing functions that can be written as weighted normal flow, i.e.

$$\bar{\mathbf{s}} = \mathbf{s} + f(s)\mathbf{n_s}, \tag{4.6}$$

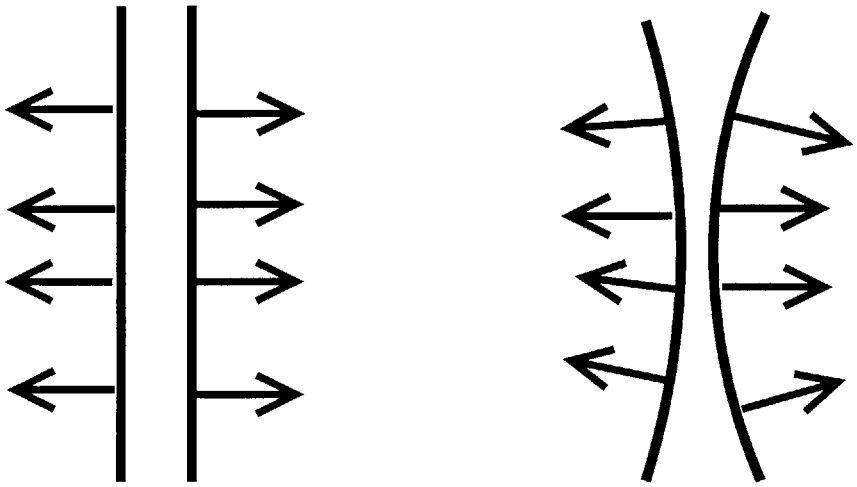
such as Fleishman et al.'s predictor, the Jacobian has n_s^T as a left eigenvector, and therefore the transpose of the adjoint has $\mathbf{n_s}$ as an eigenvector. This is under the assumption that the normal $\mathbf{n_s}$ is constant in a region around $\mathbf{s}$, i.e. the Jacobian of the normal $\nabla\mathbf{n_s}$ is $\mathbf{0}$. If higher order information about the surface at s were available, such as the curvature tensor, then this shortcoming might be addressed.

In some surface representations, such as Surfels [Pfister et al. 2000], samples store a radius for the sample as well as normal information. These radii affect the 3D bilateral filter through the area term, and as such might need to be mollified similarly to the normals. The transformation rule for radii and areas (or principal axes, in the case of elliptical samples) can also be computed via the Jacobian of the filter.

We have found that applying this scheme for normal improvement does not require any pre-mollification of the normals, and as will be discussed in the following chapter, can be used to mollify normals directly.

4.4.4 Using Normals for Weighting

It is also possible to include normals in the weighting calculation, in order to limit diffusion in cases where two surfaces pass near each other, but with opposite orientation, as in figure 4-8(a). If the samples lack connectivity information, there is no simple way to distinguish that the two surfaces are different except from normal information. If the surfaces are not treated as separate, then samples from one surface will unduly influence the denoised estimates of samples from the other. This will manifest as the



(a) (b)

Figure 4-8: Surfaces that pass near each other (a) will be pulled towards one another, even if they have differing surface normals (b).

two surfaces pulling towards one another, as in figure 4-8(b).

In order to address this, we propose the following modified bilateral function,

$$\bar{\mathbf{s}} = \frac{1}{k(\mathbf{s})} \sum_{\mathbf{p} \in X} f(|\mathbf{p} - \mathbf{s}|)\, g(|\Pi_{\mathbf{p}}(\mathbf{s}) - \mathbf{s}|)\, h(\mathbf{n_p} \cdot \mathbf{n_s})\, \Pi_{\mathbf{p}}(\mathbf{s}), \tag{4.7}$$

where $h(x)$ is a suitable, smooth function, for example,

$$h(x) = \begin{cases} 0 & \text{for} \quad x <= 0 \\ -2x^3 + 3x^2 & \text{for} \quad x > 0 \end{cases} \tag{4.8}$$

which prevents any samples with normals different by more than 90° from influencing one another. This extensions requires oriented normals, while until this point all filtering operations rely only on tangent planes without orientation. However, most models include oriented normals, so this is not a limitation in general.

Chapter 5

Results

We demonstrate the filters proposed in the previous chapter on triangulated and surfel models. We also show results for normal improvement on surfel models, and compare methods for triangle models using different normal improvement methods for mollification. Smoothing is used to denoise models corrupted with scanner noise, synthetic noise, or to remove small variations in surfaces. (The results on triangulated models are from [Jones et al. 2003], in most cases.)

5.1 Triangle Models

We first show our results for triangle models. We use the predictor and normal mollification method described in our recent work [Jones et al. 2003], rather than normal improvement via the filter. Results on a 3D scan with scanner and topological noise are shown in figure 5-1.

The necessity for mollification of the noisy normal field can be seen in figure 5-2, as well as the effect of using only a spatial falloff function. In the first case, noise is reduced, but not sufficiently well. With no influence function, details in the eye and around the lips are blurred away. The parameters controlling smoothing were $\sigma_f = 1.5$ and $\sigma_g = 0.5$, expressed in mean edge lengths.

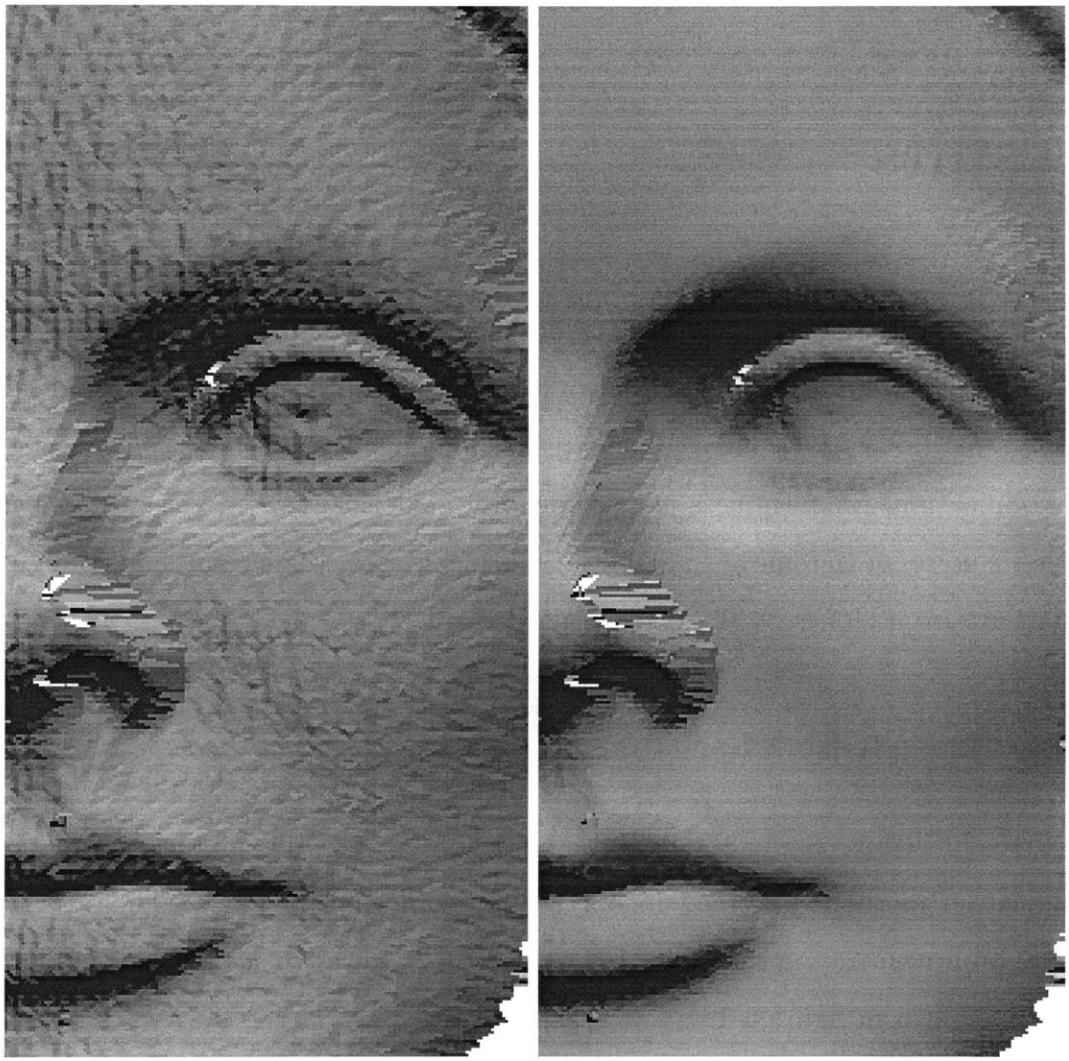


Figure 5-1: Original noisy face scan, and smoothed with the method described in [Jones et al. 2003].

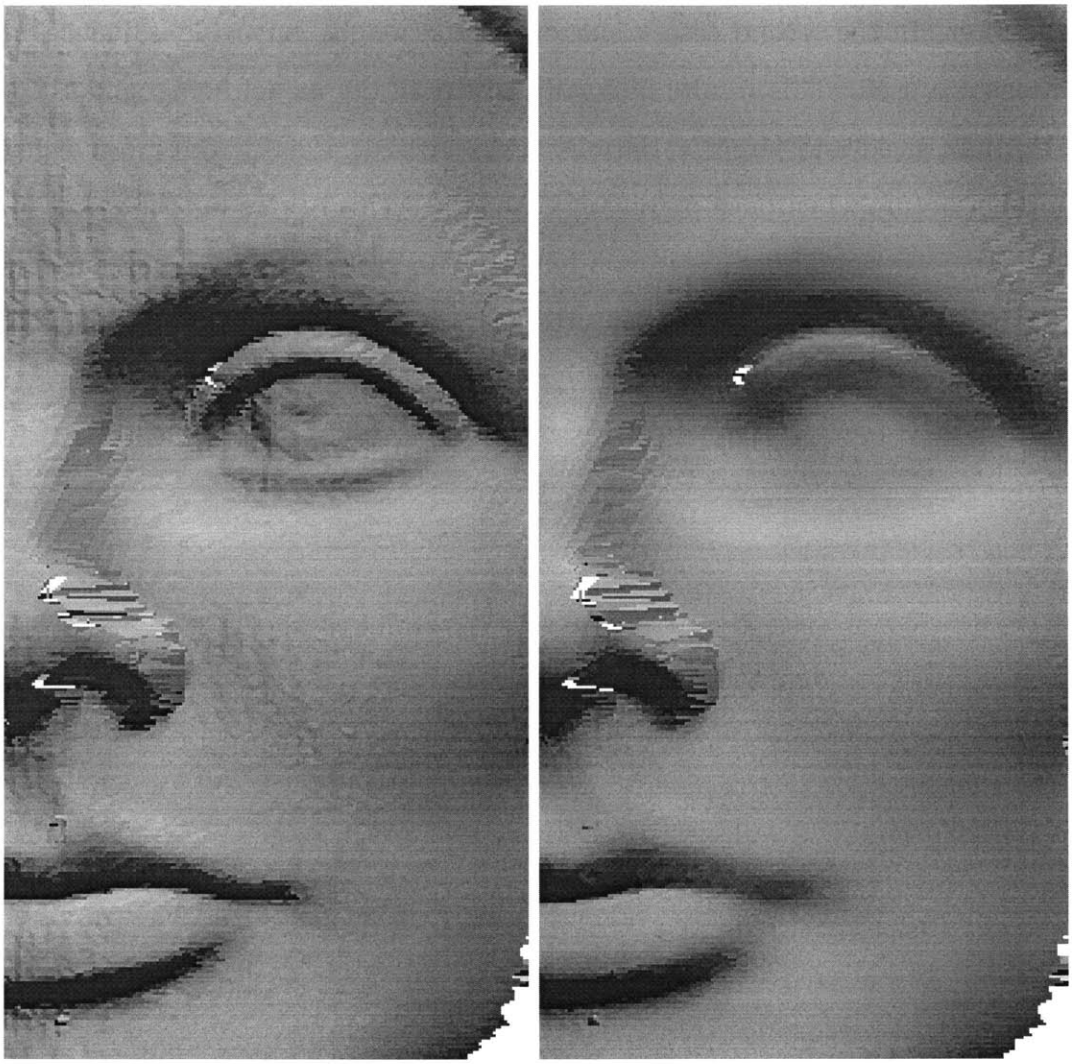


Figure 5-2: The scan from 5-1 smoothed with no mollification, and with mollification but no influence function.

To explore how smoothing is affected by the two parameters of the filter, the width of the spatial and influence weight functions, we apply the filter to a well known model with varying widths. The results are shown in figures 5-3 and 5-4. In the first example, narrow spatial and influence functions are used, as might be the case when removing noise from a model. Only the smallest variations are smoothed by the filter. In the second case, a narrow spatial weight, but wide influence weight functions are used. This results in small features of the model being smoothed out, but medium sized and larger variations are preserved. Finally, the most aggressive smoothing occurs when both parameters are large. This results in a large area being used to smooth samples, as well as a larger number of estimates being classified as "inliers". Even so, the strongest features are still preserved, such as the tips of the ears and the base around the feet.

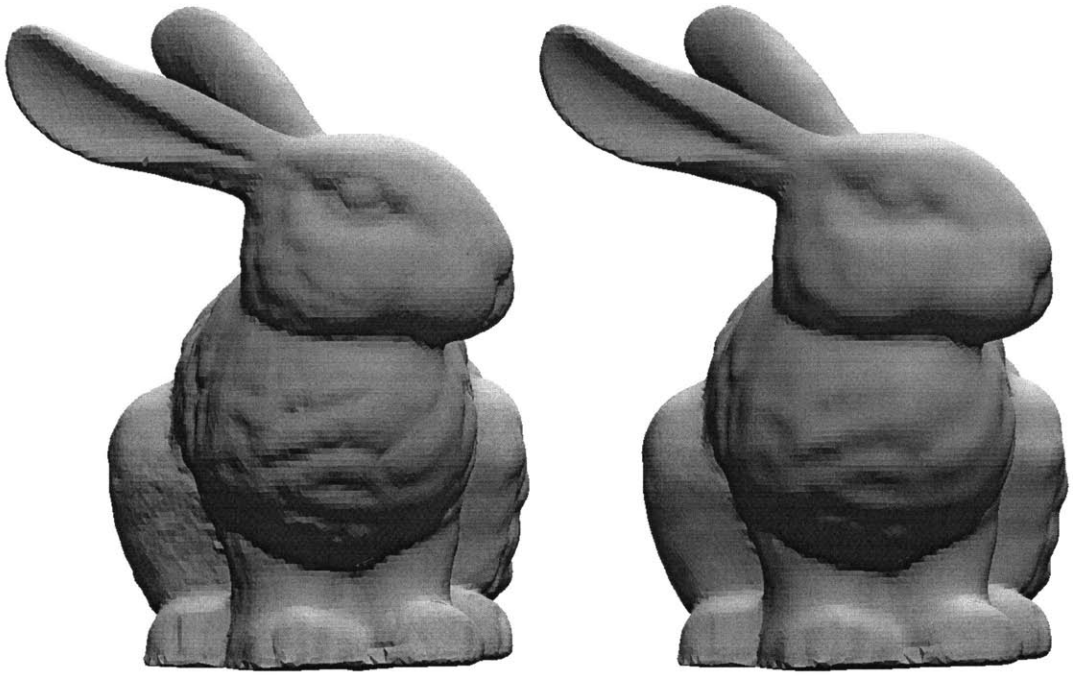


Figure 5-3: Original model, and model smoothed with narrow spatial and influence weight functions ($\sigma_f = 2,\ \sigma_g = 0.2$).

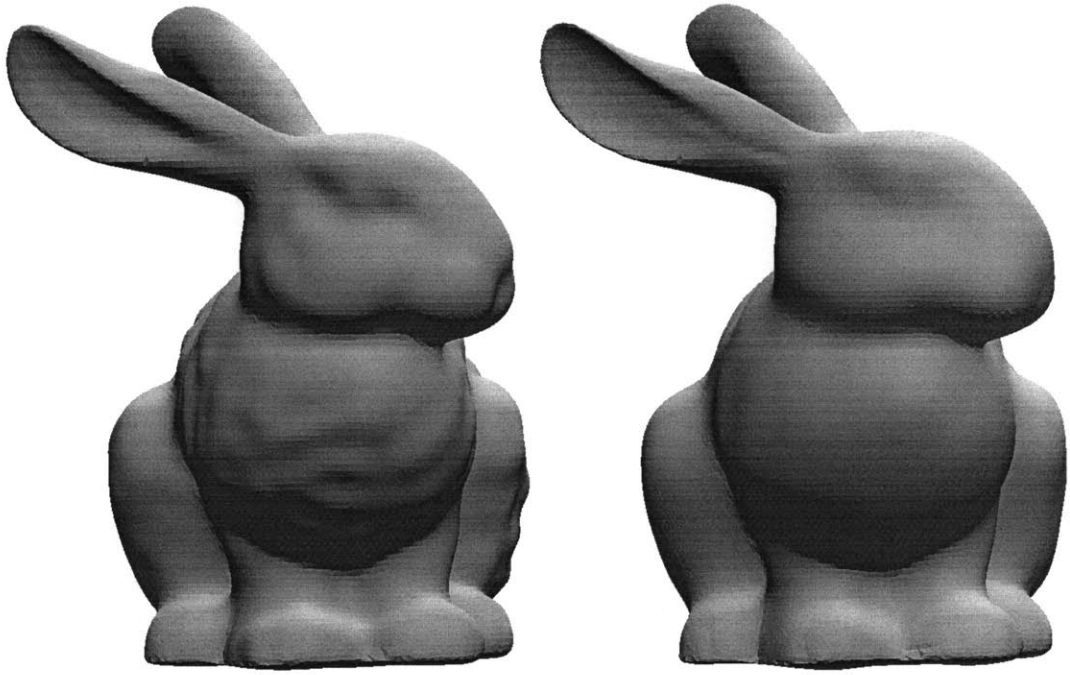


Figure 5-4: (left) Model from figure 5-3 smoothed with a narrow spatial weight function, but wide influence weight function ($\sigma_f = 2,\ \sigma_g = 4$). (right) Smoothed with wide spatial weight and influence weight functions ($\sigma_f = 4,\ \sigma_g = 4$).

Figure 5-5: Original model (courtesy of Jianbo Peng).

5.1.1 Comparison to Other Methods

We can compare our results for smoothing triangular models to other smoothing methods. In figures 5-5 through 5-7 we compare our triangle-based smoother to Wiener filtering as described by Peng et al. [2001]. Wiener filtering is controlled by a single parameter, an estimate of the variance of the noise to be removed. In figure 5-6, we compare our method to Wiener filtering under a low-noise assumption. In figure 5-7 we compare to Wiener filtering under a high-noise assumption. Our method is better at preserving features in the latter case. One benefit to Wiener filtering is that its processing time does not vary with the amount of noise to be removed. However, it does require a connected mesh with a semi-regular sampling.

In this comparison, we chose parameters for our filter to match the smoothness in substantially flat regions, and then tuned the filter to preserve features as best as possible.

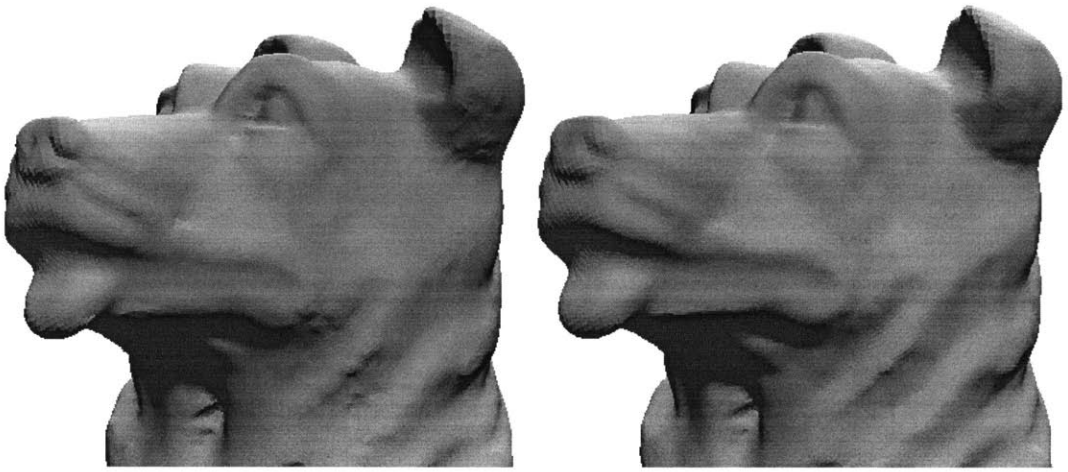


Figure 5-6: Figure 5-5 filtered with the Wiener filter of Peng et al. [2001] with a low-noise setting, and with our method ($\sigma_f = 2.7,\ \sigma_g = 0.4$).

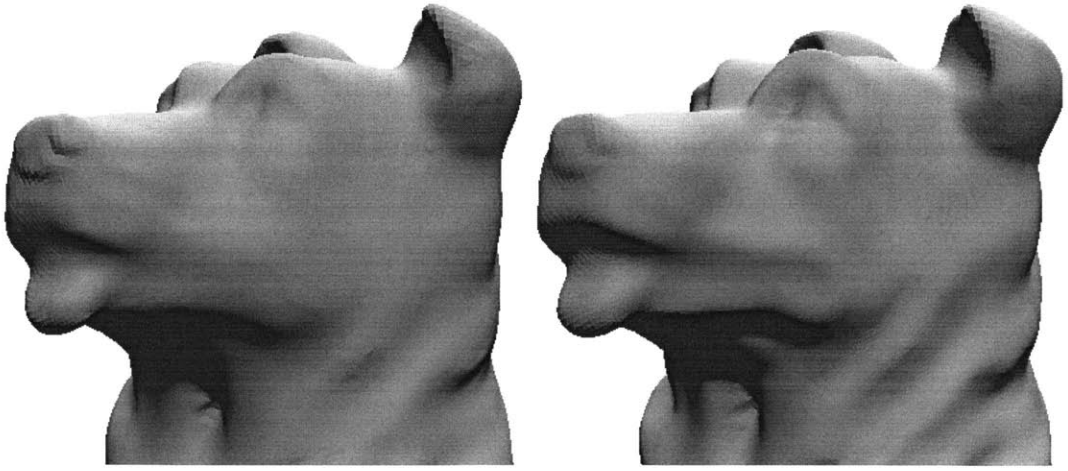


Figure 5-7: Figure 5-5 filtered with the Wiener filter of Peng et al. [2001] with a high-noise setting, and with our method ($\sigma_f = 4.0,\ \sigma_g = 1.3$).

Figure 5-8: Original model (courtesy of Martin Rumpf).

We also compare to anisotropic diffusion, in particular the method of Clarenz et al. [Clarenz et al. 2000]. Figure 5-8 shows the original model, corrupted by synthetic noise. In figure 5-9 the result of anisotropic diffusion and our method are compared. Anisotropic diffusion is able to enhance features through shock formation, while our method is not, as can be seen by comparing the results near details such as the hair. However, anisotropic diffusion relies on a connected mesh, while our method does not. As in the previous comparison, we set the parameters for our filter to first match the smoothness of flat regions, then to preserve features as best as possible.

Figure 5-9: Figure 5-8 filtered with anisotropic diffusion [Clarenz et al. 2000], and by our method ($\sigma_f = 2.5,\ \sigma_g = 1$).

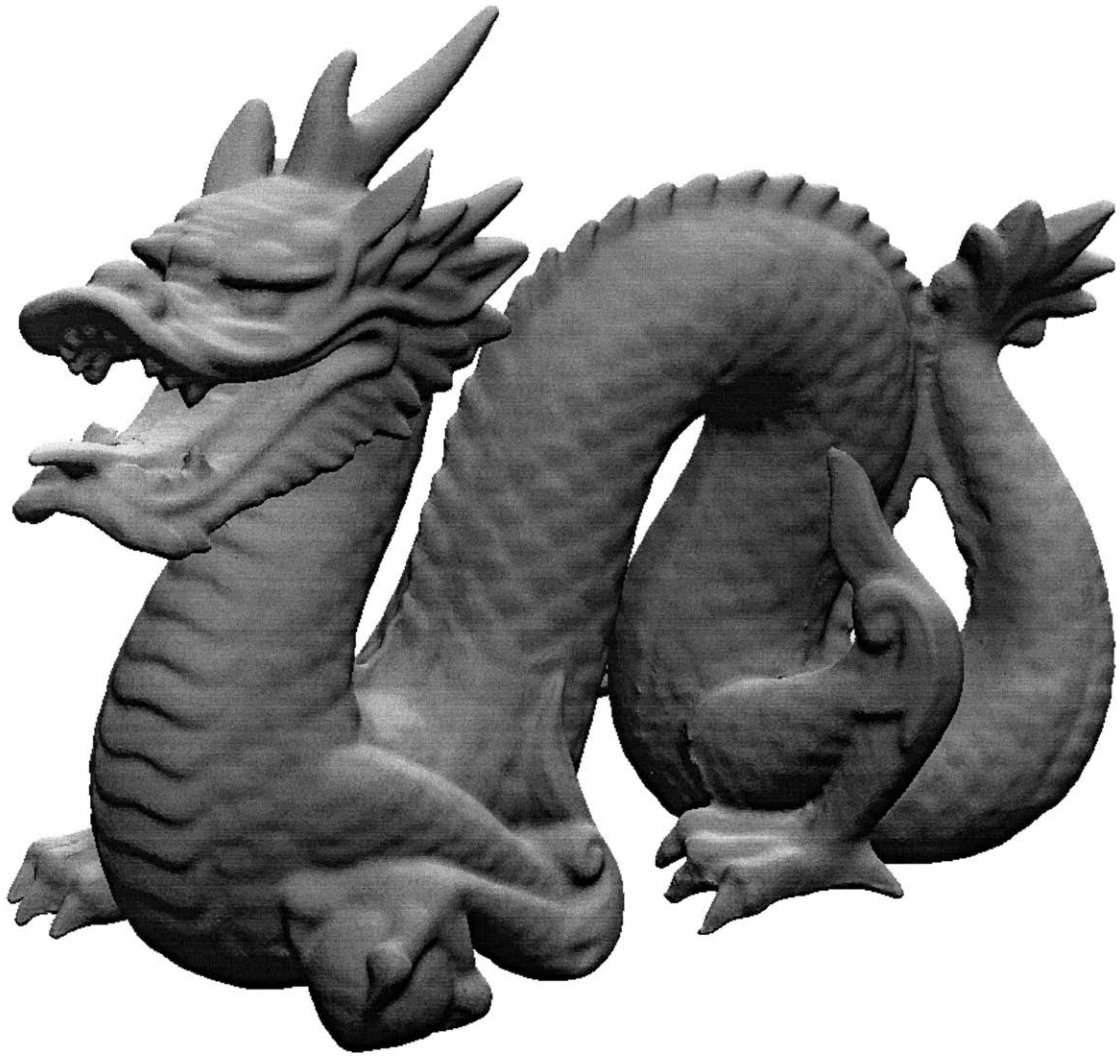


Figure 5-10: Original model (courtesy of Stanford 3D scanning repository).

We also demonstrate the stability of our algorithm, even when used to smooth all but the strongest of features on a model. Figures 5-10 and 5-11 show the dragon model smoothed with extremely wide spatial and influence weight functions. Even in this case, the filter is stable, preserving the basic shape while smoothing out most other details.

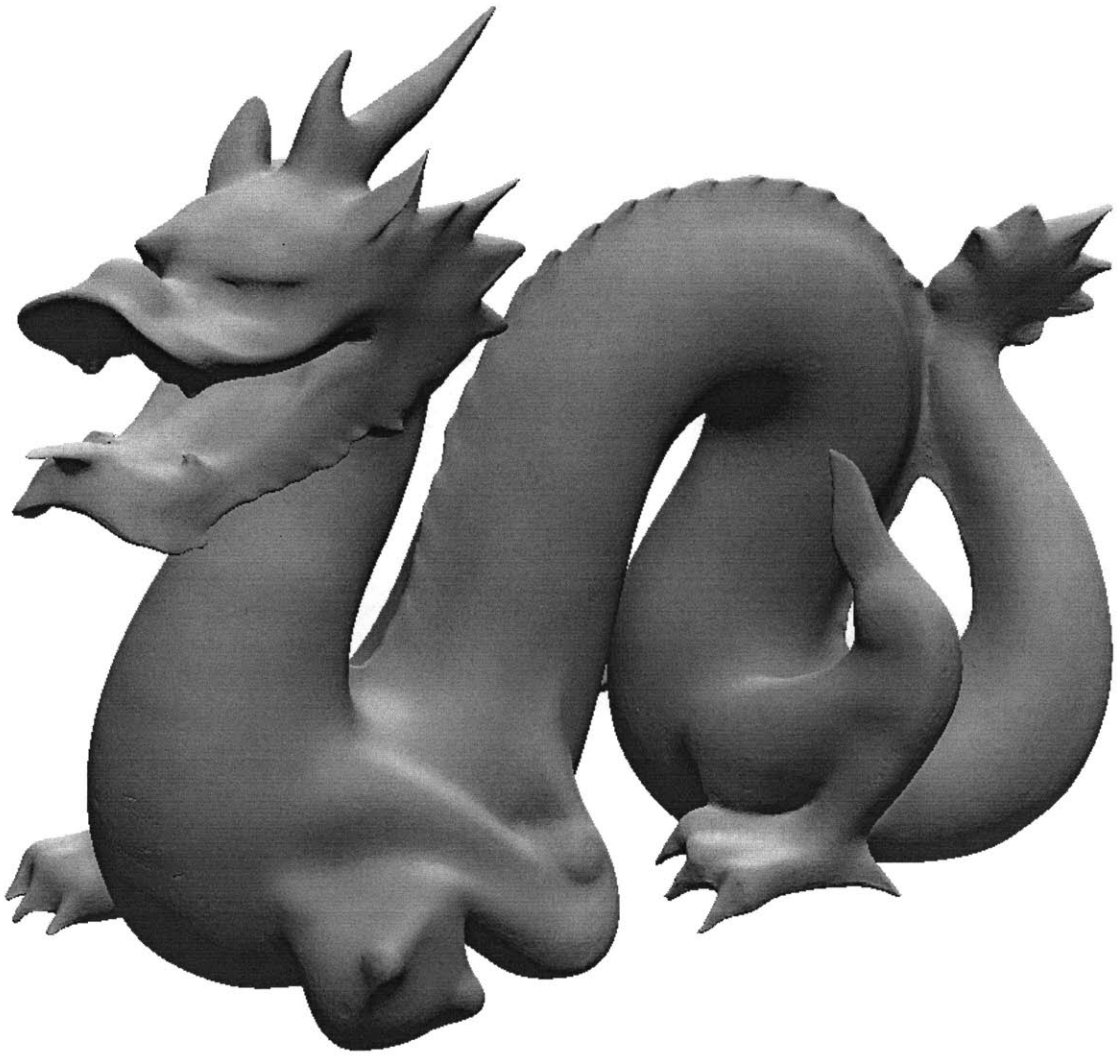


Figure 5-11: Smoothed version of figure 5-10 ($\sigma_f = 4,\ \sigma_g = 2$).

5.2 Normal Improvement

We can demonstrate the effectiveness of normal improvement using the adjoint of the filter (see section 4.4.2) on a surfel model [Pfister et al. 2000] generated from the 3D scan in figure 5-1. The appearance of surfel models depends primarily on normal orientation, rather than point position, since lighting is generally more dependent on surface orientation than position. The results of normal improvement are shown in figures 5-12 and 5-13. The smoothing parameters used were $\sigma_f = 4.0$ and $\sigma_g = 0.5$. The artifacts around the edges of the model are due to large surfels seen edge on.

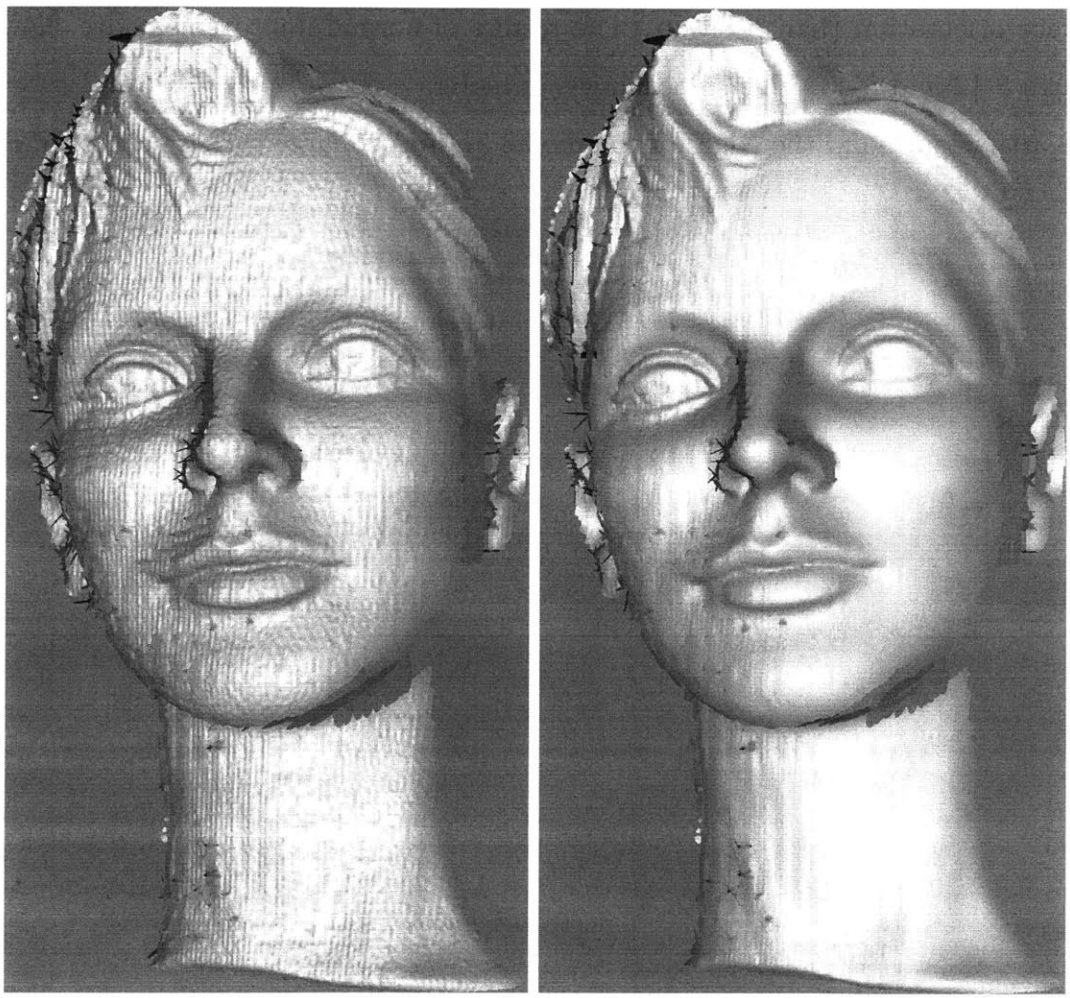


Figure 5-12: A surfel version of the scan from figure 5-1, and one iteration of normal improvement.

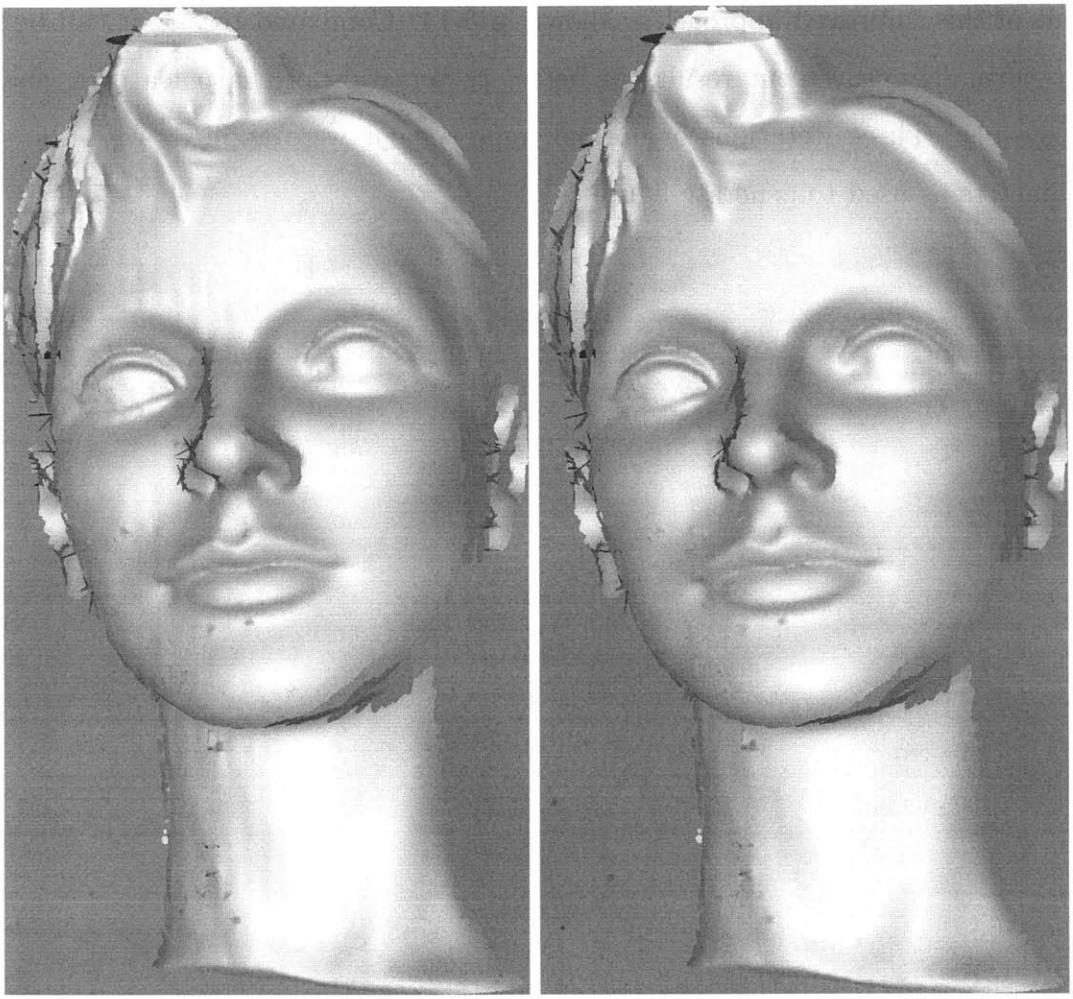


Figure 5-13: Two and three iterations of normal improvement.

5.3 Normal Improvement for Mollification

Returning to triangle models, we can use the normal improvement for mollification to improve the results from our triangle based smoother [Jones et al. 2003]. In this modification, we apply some number of normal improvement steps to the normals, then a single pass of vertex modification. For the same model as figure 5-1, the result of this improved approach is shown in 5-14. Compared to 5-1, it can be seen that normal improvement results in better preservation of sharp features, such as the eyelids. Three iterations of normal improvement were applied, with the same parameters as used to generate figure 5-1.

Figure 5-14: The result of using iterative normal improvement for mollification in conjunction with the triangle based smoother. Compare to figure 5-1 (right).

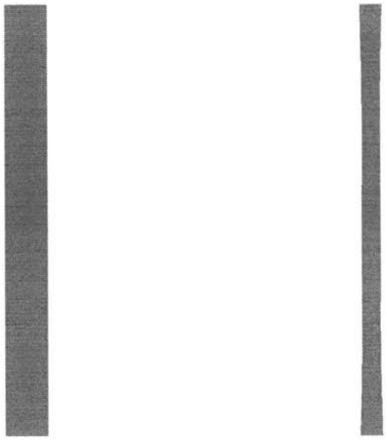


Figure 5-15: A rectangular solid, seen edge on, before and after filtering. The faces near each other are pulled towards each other and curve towards each other.

5.4 Using Normals for Weighting

Without using normal information to weight predictions, surfaces near each other will tend to pull together, even if on opposite sides of a surface. This can be seen when we filter a rectangular shape, such as in figure 5-15. The sides have been pulled closer together and curve towards each other.

If we use normal information to weight predictions, as discussed in section 4.4.4, the shape is unchanged. Normal information also results in better denoising. We apply the triangle-based filter with feature preserving mollification to a noisy version of the fandisk model. The noisy mesh, and the result of smoothing it with our triangle-based method (with the original mollification method) is shown in 5-16. The results of smoothing with and without normal information are shown in figure 5-17. The most noticeable changes are on the small, frontmost face at its most narrow points, and near the edges and corners of the mesh. The change is small, but subtle, possibly because the result of smoothing without normal weighting is so close to the original model that there is not much room for improvement. It is also possible that the sampling for this model is too coarse for good estimates of the normal field to be formed from the original, noisy samples.

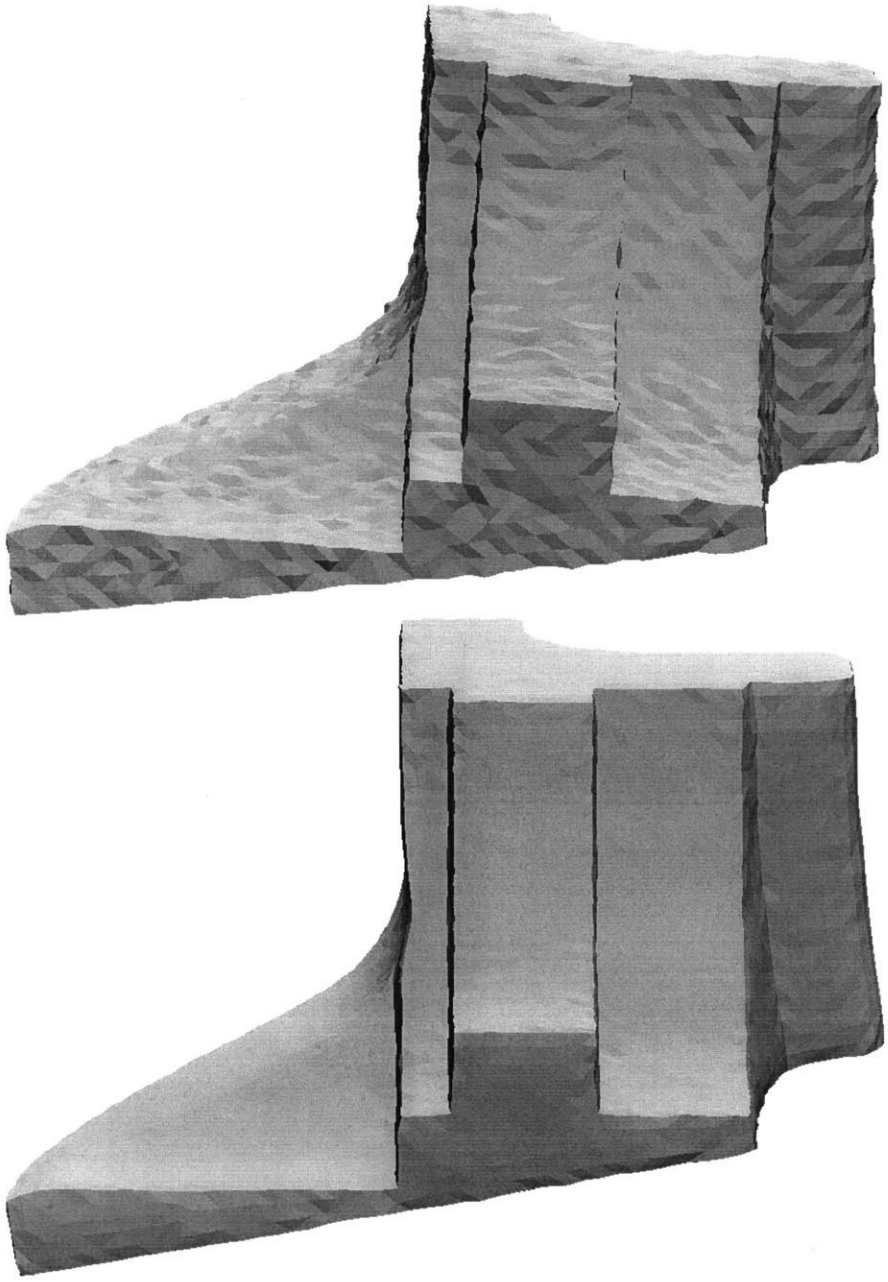


Figure 5-16: A noisy version of the fandisk model, and the result of smoothing with our triangle based smoother.

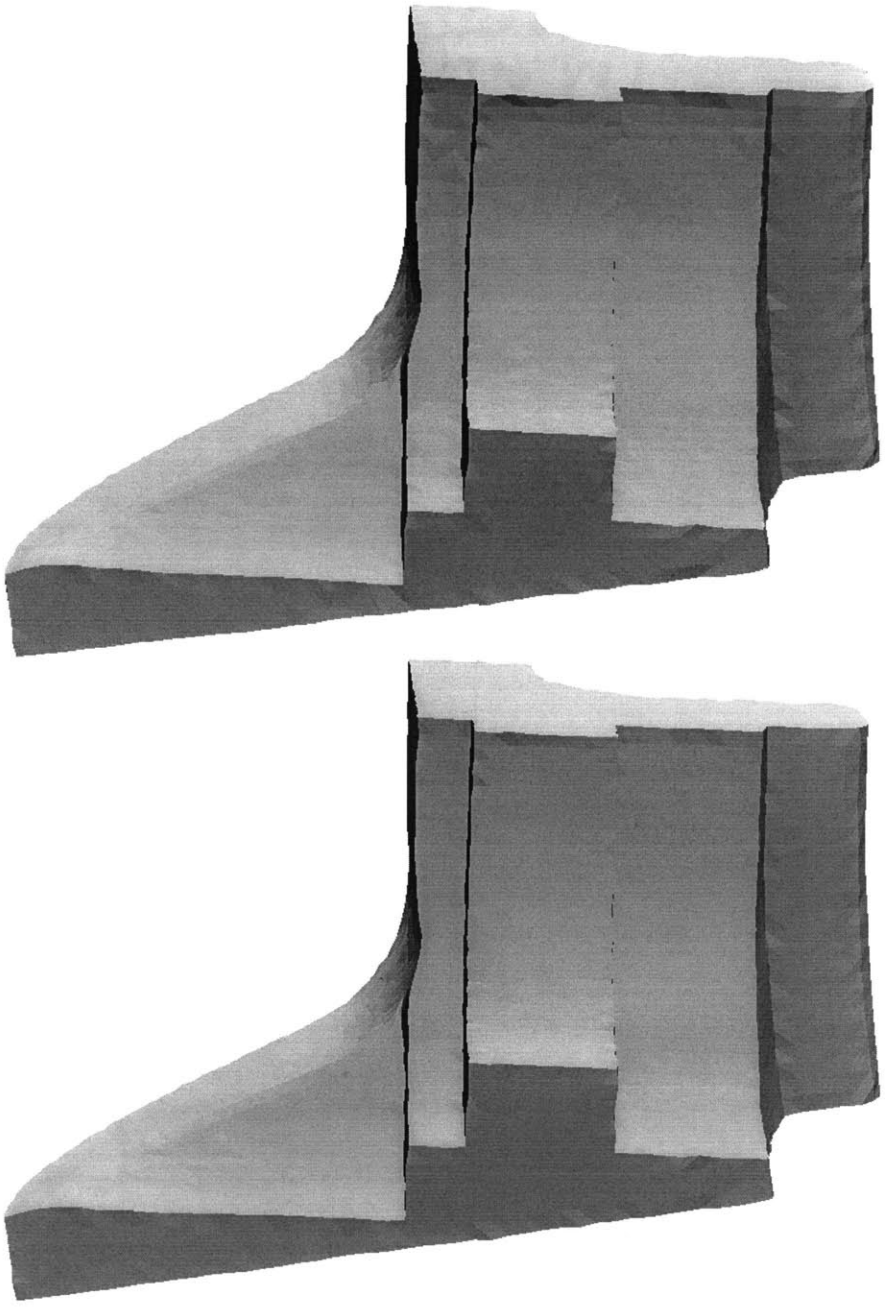


Figure 5-17: Smoothing the mesh from figure 5-16 without taking into account normal information in weighting predictions, and with normal information used in weighting. Note the differences in the frontmost face, particularly at its most narrow point.

Chapter 6

Conclusions

We have demonstrated feature preserving filters for 3D shapes based on robust statistics and estimation. In particular, we have extended the bilateral filter from the domain of images to 3D shapes. Our methods are applicable to a wide variety of shape representations, including triangle meshes, triangle soups, and point clouds.

In order to extend the bilateral filter from images to 3D shapes, we first consider it as a weighted combination of *predictors*, rather than simply values, as in most image filters. We then use first order predictors to separate position and signal on the shape. This allows us to generalize the bilateral filter to 3D shapes in a straightforward way.

We have also introduced methods for smoothing the normal fields of surfaces in any of the above representations, by treating our filter as a free-form deformation and considering how it affects differential entities such as normals. We have used this feature preserving normal smoothing for mollification of normal fields, improving the performance of our filter when applied to sample positions.

Our methods are effective for noise removal and mesh smoothing, preserving the strongest features while removing noise and small surface variations. We have also demonstrated treating the filter as a free-form deformation, leading to a very effective method for improving normals. This normal improvement can also be used as a mollification method, to further improve estimation of vertex positions.

In the future, we would like to explore extensions to volume data. We would also like to explore how the 3D bilateral filter could be used on shapes with color infor-

mation, such as textures, to smooth both the shape and the texture while respecting and preserving the features in both. More generally, we would like to explore the possibility of building a *scale space* [Weickert 1996] for meshes.

Bibliography

[Alexa et al. 2001] ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. In *IEEE Visualization 2001*, 21–28.

[Alexa et al. 2003] ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2003. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9*, 1, 3–15.

[Alexa 2002] ALEXA, M. 2002. Wiener Filtering of Meshes. In *Proceedings of Shape Modeling International*, 51–57.

[Amenta et al. 1998] AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998. A New Voronoi-Based Surface Reconstruction Algorithm. In *Proceedings of SIGGRAPH 98*, ACM SIGGRAPH / Addison Wesley, Orlando, Florida, Computer Graphics Proceedings, Annual Conference Series, 415–422. ISBN 0-89791-999-8.

[Bajaj and Xu 2003] BAJAJ, C., AND XU, G. 2003. Anisotropic Diffusion on Surfaces and Functions on Surfaces. *ACM Trans. Gr. 22*, 1, 4–32.

[Barash 2001] BARASH, D. 2001. Bilateral Filtering and Anisotropic Diffusion: Towards a Unified Viewpoint Danny Barash. In *Scale-Space 2001*, 273–280.

[Barash 2002] BARASH, D. 2002. A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing and the Nonlinear Diffusion Equation. *IEEE PAMI 24*, 6, 844.

[Barr 1984] BARR, A. H. 1984. Global and Local Deformations of Solid Primitives. In *Proceedings of SIGGRAPH*, 21–30.

[Belyaev and Ohtake 2001] BELYAEV, A., AND OHTAKE, Y. 2001. Nonlinear Diffusion of Normals for Crease Enhancement. In *Vision Geometry X, SPIE Annual Meeting*, 42–47.

[Bernardini and Bajaj 1997] BERNARDINI, F., AND BAJAJ, C. L. 1997. Sampling and Reconstructing Manifolds using Alpha-Shapes. In *Proc. 9th Canadian Conf. Computational Geometry*, 193–198.

[Black et al. 1998] BLACK, M., SAPIRO, G., MARIMONT, D., AND HEEGER, D. 1998. Robust anisotropic diffusion. *IEEE Trans. Image Processing 7*, 3, 421–432.

[Carr et al. 2001] CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and Representation of 3D Objects With Radial Basis Functions. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 67–76. ISBN 1-58113-292-1.

[Choudhury and Tumblin 2003] CHOUDHURY, P., AND TUMBLIN, J. 2003. The Trilateral Filter for High Contrast Images and Meshes. *Proc. of the Eurographics Symposium on Rendering*, 186–196.

[Clarenz et al. 2000] CLARENZ, U., DIEWALD, U., AND RUMPF, M. 2000. Anisotropic geometric diffusion in surface processing. In *IEEE Visualization 2000*, 397–405.

[Curless and Levoy 1996] CURLESS, B., AND LEVOY, M. 1996. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of SIGGRAPH 96*, 303–312.

[Desbrun et al. 1999] DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *Proceedings of SIGGRAPH 99*, 317–324.

[Desbrun et al. 2000] DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 2000. Anisotropic Feature-Preserving Denoising of Height Fields and Bivariate Data. In *Graphics Interface*, 145–152.

[Durand and Dorsey 2002] DURAND, F., AND DORSEY, J. 2002. Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. *ACM Trans. Gr. 21*, 3, 257–266.

[Elad 2002] ELAD, M. 2002. On the Bilateral Filter and Ways to Improve It. *IEEE Trans. on Image Processing 11*, 10 (Oct.), 1141–1151.

[Fleishman et al. 2003] FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral Mesh Denoising. *ACM Trans. Gr. (Proceedings of ACM SIGGRAPH)*, 950–953.

[Fourier 1955] FOURIER, J. 1955. *The Analytical Theory of Heat.* Dover Publications, Inc., New York. Replication of the English translation that first appeared in 1878. Original work: "Theories Analytique de la Chaleur", Paris, 1822.

[Guskov and Wood 2001] GUSKOV, I., AND WOOD, Z. 2001. Topological Noise Removal. In *Graphics Interface 2001*, 19–26.

[Hampel et al. 1986] HAMPEL, F. R., RONCHETTI, E. M., ROUSSEEUW, P. J., AND STAHEL, W. A. 1986. *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons. ISBN 0471-63238-4.

[Huber 1981] HUBER, P. J. 1981. *Robust Statistics.* John Wiley and Sons.

[Jones et al. 2003] JONES, T. R., DURAND, F., AND DESBRUN, M. 2003. Non-Iterative, Feature-Preserving Mesh Smoothing. *ACM Transactions on Graphics*, 943–949.

[Levin 2001] LEVIN, D. 2001. Mesh-independent surface interpolation. In *Advances in Computational Mathematics.*

[Levoy et al. 2000] LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Proceedings of SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 131–144. ISBN 1-58113-208-5.

[Meyer et al. 2002] MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2002. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Proceedings of Visualization and Mathematics.*

[Mihçak et al. 1999] MIHÇAK, M., KOZINETSEV, I., RAMCHANDRAN, K., AND MOULIN, P. 1999. Low-complexity image denois-

ing based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters 6*, 12, 300–303.

[Moulin and Liu 1999] MOULIN, P., AND LIU, J. 1999. Analysis of Multiresolution Image Denoising Schemes Using Generalized-Gaussian and Complexity Priors. *IEEE Transactions on Information Theory*, 909–919.

[Murio 1993] MURIO, D. A. 1993. *The mollification method and the numerical solution of ill-posed problems.* Wiley.

[Ohtake et al. 2002] OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2002. Mesh Smoothing by Adaptive and Anisotropic Gaussian Filter Applied to Mesh Normal. In *Vision, modeling and visualization*, 203–210.

[Pauly and Gross 2001] PAULY, M., AND GROSS, M. 2001. Spectral Processing of Point-Sampled Geometry. In *Proceedings of ACM SIGGRAPH 2001*, 379–386.

[Peng et al. 2001] PENG, J., STRELA, V., AND ZORIN, D. 2001. A Simple Algorithm for Surface Denoising. In *Proceedings of IEEE Visualization 2001*, 107–112.

[Perona and Malik 1990] PERONA, P., AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE PAMI 12*, 7, 629–639.

[Pfister et al. 2000] PFISTER, H., ZWICKER, M., VAN BAAR, J., AND GROSS, M. 2000. Surfels: Surface Elements as Rendering Primitives. In *Siggraph 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, K. Akeley, Ed., 335–342.

[Preusser and Rumpf 2002] PREUSSER, T., AND RUMPF, M. 2002. A Level Set Method for Anisotropic Geometric Diffusion in 3D Image Processing. *SCIAM*, 1772–1793.

[Rusinkiewicz et al. 2002] RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-Time 3D Model Acquisition. *ACM Trans. Gr. 21*, 3 (July), 438–446.

[Smith and Brady 1997] SMITH, S. M., AND BRADY, J. M. 1997. SUSAN - a new approach to low level image processing. *IJCV 23*, 45–78.

[Soucy and Laurendeau 1992] SOUCY, M., AND LAURENDEAU, D. 1992. Surface Modeling from Dynamic Integration of Multiple Range Views. In *International Conference on Pattern Recognition*, I:449–452.

[Strang 1986] STRANG, G. 1986. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press.

[Strela et al. 2000] STRELA, V., PORTILLA, J., AND SIMONCELLI, E. 2000. Image Denoising Using a Local Gaussian Scale Mixture Model in the Wavelet Domain. In *SPIE 45th Annual Meeting.*

[Tasdizen et al. 2002] TASDIZEN, T., WHITAKER, R., BURCHARD, P., AND OSHER, S. 2002. Geometric Surface Smoothing via Anisotropic Diffusion of Normals. In *Proceedings, IEEE Visualization 2002*, 125–132.

[Taubin 1995] TAUBIN, G. 1995. A Signal Processing Approach to Fair Surface Design. In *Proceedings of SIGGRAPH 95*, 351–358.

[Taubin 2001] TAUBIN, G. 2001. Linear Anisotropic Mesh Filtering. Tech. Rep. IBM Research Report RC2213.

[Tomasi and Manduchi 1998] TOMASI, C., AND MANDUCHI, R. 1998. Bilateral Filtering for Gray and Color Images. In *Proc. IEEE Int. Conf. on Computer Vision*, 836–846.

[Turk and Levoy 1994] TURK, G., AND LEVOY, M. 1994. Zippered Polygon Meshes from Range Images. In *Proceedings of SIGGRAPH 94*, 311–318.

[Šrámek and Kaufman 1999] ŠRÁMEK, M., AND KAUFMAN, A. E. 1999. Alias-Free Voxelization of Geometric Objects. *IEEE Transactions on Visualization and Computer Graphics 5*, 3, 251–267.

[Weickert 1996] WEICKERT, J. 1996. *Anisotropic diffusion in image processing.* PhD thesis, Dept. of Mathematics, University of Kaiserslautern, Germany.

[Wheeler et al. 1998] WHEELER, M., SATO, Y., AND IKEUCHI, K. 1998. Consensus surfaces for modeling 3D objects from multiple range images. In *Proceedings of ICCV '98*, 917 – 924.

[Wood et al. 2002] WOOD, Z., HOPPE, H., DESBRUN, M., AND SCHRÖDER, P. 2002. Isosurface Topology Simplification. http://www.multires.caltech.edu/pubs/.

[Yagou et al. 2003] YAGOU, H., OHTAKE, Y., AND BELYAEV, A. 2003. Mesh Denoising via Iterative Alpha-Trimming and Nonlinear Diffusion of Normals with Automatic Thresholding. In *Proceedings of Computer Graphics International*, 28–33.

[You et al. 1996] YOU, Y., XU, W., TANNENBAUM, E., AND KAVEH, M. 1996. Behavioral analysis of anisotropic diffusion in image enhancement. *IEEE Trans. on Image Processing 5*, 11, 1539–1553.

[Zorin et al. 1997] ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. 1997. Interactive Multiresolution Mesh Editing. In *Proceedings of SIGGRAPH 97*, 259–268.