Feature-preserving Non-local Denoising of Static and Time-varying Range Data

Oliver Schall Alexander Belyaev
Hans-Peter Seidel

MPI–I–2006–4–007 June 2006

Authors' Addresses

Oliver Schall
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Alexander Belyaev
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Abstract

We present a novel algorithm for accurately denoising static and time-varying range data. Our approach is inspired by similarity-based non-local image filtering. We show that our proposed method is easy to implement and outperforms recent state-of-the-art filtering approaches. Furthermore, it preserves fine shape features and produces an accurate smoothing result in the spatial and along the time domain.

Contents

1 Introduction

With the increasing usage of scanning devices, denoising of digitized models became one of the most fundamental problems in computer graphics. It remains a challenging task to remove the inevitable noise created in every acquisition process while preserving the details of the underlying shape. Especially, fine features are often lost if no special treatment is provided. Therefore, a large variety of smoothing algorithms has been introduced in recent years in the fields of image processing, computer vision, and computer graphics. While many image processing approaches are also specifically designed for video denoising, methods for 3-dimensional data are usually applied to static point clouds or meshes.

Recent improvements in scanning technology [24, 25] now permit the acquisition of time-varying range data. This is usually accomplished by projecting structured light patterns on the scanned object. Then range data is computed off-line from the recorded frames using triangulation algorithms from computer vision.

In this paper, we propose a method for denoising this new type of data which is based on non-local image filtering [5]. The main idea of this filter is to determine the denoised pixel intensity as a weighted average of similar pixel intensities in its vicinity. The similarity of two pixels is determined by comparing their local neighborhoods. We extend this approach to static and dynamic range data and show that our algorithm accurately preserves fine shape features, is easy to implement and is able to outperform recent state-of-the-art filtering approaches.

To our knowledge, our approach is the first which is designed to denoise time-varying geometric data. We believe that denoising this new type of data is the first step to open it for a wide use in various fields of computer graphics.

2 Previous Work

Recent state-of-the-art approaches in image denoising comprise, for instance, the well known bilateral filter [22] and its recently proposed extension to non-local neighborhoods [5]. Paris and Durand [17] introduced an interesting acceleration technique for the bilateral filter which allows it to be mostly expressed as simple linear convolutions. Other works adapt neighborhood filters for video processing [4, 15].

Many mesh denoising methods are derived from signal or image processing approaches. Taubin [21] first introduced signal processing on meshes based on the definition of the Laplacian operator on surfaces. Desbrun et al. [6] proposed a geometric diffusion algorithm for irregular meshes and introduced the use of an implicit integration method to stabilize the flow and to allow larger time steps. In [1], Wiener filtering is applied to meshes. Hildebrandt and Polthier [10] introduced a new variant of anisotropic mean curvature flow which preserves non-linear features. Fleishman et al. [9] proposed an anisotropic mesh denoising algorithm derived from the bilateral neighborhood filter for images. Concurrently, Jones et al. [12] introduced a similar method based on robust statistics and local first-order predictors of a surface. Recent work of Yoshizawa et al. [23] extends the non-local image filter to meshes by computing a local RBF approximation to define the similarity measure. Our work proposes a different similarity measure for range scans which does not require to compute a local approximation and thus allows a faster evaluation. Furthermore, we introduce how to process time-varying range data.

Denoising can be either applied before or after meshing acquired point clouds. Filtering of meshes is usually faster since the given connectivity permits an efficient access to neighboring samples. On the other hand, surface reconstruction from noisy point clouds is a difficult task which makes prior denoising desirable. Pauly and Gross [18] create a spectral decomposition of a point cloud and denoise it by manipulation of the spectral coefficients. Lange and Polthier [13] denoise point clouds using anisotropic mean curvature flow.

An important group of algorithms that recently attracted the interest of many researchers are moving-least squares (MLS) approaches. They were first proposed by Levin [14] and introduced to computer graphics by Alexa et al. [2]. The main idea of MLS is the definition of a projection operator which takes points scattered in the vicinity of a surface onto the surface itself. More precisely, the MLS surface is defined by the fixpoints of the given projection operator. Mederos et al. [16] applied the MLS projection for point cloud denoising. Amenta and Kil [3] analyzed different MLS operators by separating them into two components. They used this representation to introduce a new variant of MLS with a better behavior near sharp features. Fleishman et al. [8] represented sharp features by defining piecewise smooth moving least-squares surfaces using a method from robust statistics. Dey and Sun [7] recently proposed the AMLS operator which provides reconstruction guarantees for the underlying surface for a point set with a non-uniform sampling density.

Other algorithms are based on statistical data analysis. Pauly et al. [19] introduced a framework for measuring uncertainty in point-sampled geometry which can be used, for instance, for merging range scans. Schall et al. [20] used locally defined kernels to define a global probability distribution function. Point positions on a smooth surface are then found by moving the noisy point set to maximum likelihood positions. Recently, [11] proposed how to produce a smooth point cloud from a given one using Bayesian statistics.

3 Non-local Denoising

We begin this section by describing the idea of non-local filtering for images in more detail before we introduce our extension of this approach for denoising static range data. Building on this extension, we then show how to apply our algorithm to filter time-varying range data.

3.1 Non-local Image Filtering

The non-local image filter [5] belongs to the group of neighborhood filtering schemes, which define the intensity value of a restored pixel of an image as the weighted average of neighboring pixels with similar intensity values.

More precisely, if an image $\mathcal{I} = \{I(\mathbf{u}) | \mathbf{u} \in P\}$ is given, where $\mathbf{u} = (x, y)$ is a pixel and $I(\mathbf{u})$ is the intensity value at $\mathbf{u}$, the smoothed pixel intensity $I'(\mathbf{u})$ can be computed as the average of all pixel intensities in the image

$$I'(\mathbf{u}) = \frac{\sum_{\mathbf{v} \in P} \Phi(\mathbf{u}, \mathbf{v}) \cdot I(\mathbf{v})}{\sum_{\mathbf{v} \in P} \Phi(\mathbf{u}, \mathbf{v})}$$

weighted by a similarity factor which measures the similarity between $\mathbf{u}$ and $\mathbf{v}$ as

$$\Phi(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\sum_{\mathbf{o}} G_a(||\mathbf{o}||) \cdot |I(\mathbf{u} + \mathbf{o}) - I(\mathbf{v} + \mathbf{o})|^2}{h^2} \right).$$

Figure 3.1 illustrates the computation of the similarity measure. It depends on the pixel-wise intensity difference of two square neighborhoods centered at the pixels $\mathbf{u}$ and $\mathbf{v}$. The vector $\mathbf{o}$ denotes the offset between the center pixel and an arbitrary neighborhood pixel. The influence of a pixel pair on the similarity falls with increasing Euclidean distance to the center of the neighborhoods. For the distance weighting a Gaussian kernel $G_a(\cdot)$ with a user-defined standard deviation a is used. Additionally, the method depends on the parameter h which controls the degree of smoothing.

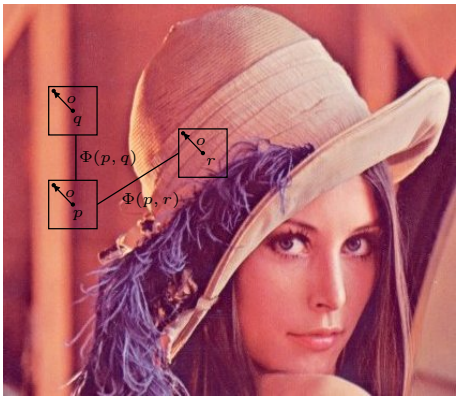

Figure 3.1: The similarity of neighborhoods is computed based on the pixel-wise difference of intensity values. Similar neighborhoods of p and q have a large weight $\Phi(p, q)$, while different neighborhoods of p and r have a small weight $\Phi(p, r)$.

3.2 Static Range Data

We want to adapt this approach from the 2-dimensional plane to range data. This is not a straight-forward task as image pixels are usually aligned on a regular and equispaced grid which is in general not true for range images. Its reason is that an acquisition device measures the distance between itself and the object along its line-of-sight which is not the depth in the form of a height field. Additionally, computing height data from the given depths causes that the sampling positions are no longer equispaced. Hence, our problem is different from image denoising, where pixels are usually aligned on a regular and equispaced grid. We assume that the data is given in the form of data points $\mathbf{p}_i$ which are arranged on a regular grid structure. In this way, the neighborhood information for all points is known, but they are not required to be equispaced. Since this data representation can be easily computed from the output of different scanning devices, our algorithm is easily applicable to filter different types of range data.

Similar to the bilateral filtering algorithm [9], we find counterparts for grey values of an image by the heights of points over a tangent plane computed at a given point. By determining the weighted average of these offsets, we find the displacement for the point to remove the additive noise component from the range scan surface. For this, we first estimate normals $\mathbf{n}_i$ for all points $\mathbf{p}_i$ by least-squares fitting to their one-ring neighborhood. Although the resulting normals are noisy, no mollification in contrast to [12] is necessary to apply our algorithm. After this, we determine the filtered points $\mathbf{p}'_i$

by computing

$$\mathbf{p}_i' = \mathbf{p}_i - \frac{\sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i)} \Phi_d \cdot \Phi_s \cdot [(\mathbf{p}_i - \mathbf{p}_j) \cdot \mathbf{n}_i] \cdot \mathbf{n}_i}{\sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i)} \Phi_d \cdot \Phi_s}$$

where Φ_d represents the distance and Φ_s the similarity weight.

Unlike the non-local image filtering algorithm, we do not sum over all point positions to filter a point but over a local square neighborhood $\mathcal{N}(\mathbf{p}_i)$ surrounding $\mathbf{p}_i$. Additionally, we separate the distance weighting factor from the similarity measure. This allows us a more efficient computation of the similarity weight and the denoised point position $\mathbf{p}_i'$.

The fundamental difference between our method and bilateral filtering is the selection of the similarity weight Φ_s. Unlike the bilateral filtering algorithm, where Φ_s only weights the similarity between the two points $\mathbf{p}_i$ and $\mathbf{p}_j$, our approach considers the similarity of their geometric neighborhoods:

$$\Phi_d(\mathbf{p}_i, \mathbf{p}_j) = e^{-\frac{||\mathbf{p}_i - \mathbf{p}_j||^2}{d^2}} \qquad \Phi_s(\mathbf{p}_i, \mathbf{p}_j) = e^{-\frac{Sim(\mathbf{p}_i, \mathbf{p}_j)^2}{s^2}}$$

$$Sim(\mathbf{p}_i, \mathbf{p}_j) = \frac{\sum_{o \in O} |(\mathbf{p}_{i+o} - \mathbf{p}_{j+o}) \cdot \mathbf{n}_i|^2}{||O||}.$$

This results in a better and more homogeneous filtering performance. We compute the point-wise difference of two square neighborhoods centered at $\mathbf{p}_i$ and $\mathbf{p}_j$ and project the distances onto the normal $\mathbf{n}_i$. This gives us the point-wise height difference of both neighborhoods which is averaged to compute the similarity $Sim(\mathbf{p}_i, \mathbf{p}_j)$. We use Gaussian weighting functions for Φ_d and Φ_s and an automatic procedure to determine their bandwidths d and s. For this, we first choose random points $\mathbf{p}_k$ of the range scan. We then determine the maximal distance of the points of $\mathcal{N}(\mathbf{p}_k)$ to $\mathbf{p}_k$ and the standard deviation of all offsets to the tangent plane defined at $\mathbf{p}_k$. The average maximal distance and standard deviation over all random samples are then assigned μ_d and σ_s. We set $d = 0.75\mu_d$ and $s = \sigma_s$.

The user-defined parameters of our algorithm are thus the size of the neighborhood $\mathcal{N}(\mathbf{p}_i)$ which controls the degree of smoothing and the size of the neighborhood used to determine $Sim(\mathbf{p}_i, \mathbf{p}_j)$ which regulates the homogeneity of the filtering result.

Similar to other neighborhood filtering schemes for meshes our algorithm shrinks the object. This problem can be corrected in a post-processing step using for instance a volume preservation technique [6]. On the other hand, our method does not require special boundary treatment which is important as scanned data is not closed and often has holes. Furthermore, our algorithm only changes the point position in normal direction. This avoids point-drifts which would introduce irregularities in the scan.

3.3 Time-varying Range Data

Building on the previous section, we now propose how to extend our algorithm to handle time-varying range data. The data is given as a sequence of frames each of which is a static range scan. When we apply our algorithm to each frame independently, we obtain a result that is satisfying for each frame but which is not temporally stable.

Therefore, we extend $\mathcal{N}(\mathbf{p}_i)$ which is only defined as a spatial neighborhood in the previous section by the temporal domain. This means we choose for $\mathcal{N}(\mathbf{p}_i)$ sample points not only inside the current frame but also in neighboring frames. We usually consider one frame before and after the current frame for $\mathcal{N}(\mathbf{p}_i)$. In the following, we use the notation $\mathcal{N}_k(\mathbf{p}_i)$ for the slice of the neighborhood $\mathcal{N}(\mathbf{p}_i)$ which is contributed by the frame k. Consequently, we have to adapt Φ_d and Φ_s to weight the distance and the similarity between $\mathbf{p}_i$ and $\mathbf{p}_j$ which can be a points in different frames. We adapt the bandwidths of Φ_d and Φ_s depending on the frame $\mathbf{p}_j$ is associated with. We detect the parameters automatically as described in Sec. 3.2 for each frame k and identify the weighting functions as Φ_{dk} and Φ_{sk}.

Similar to the spatial domain, we want that neighborhoods from distant frames contribute less to the new point position. We therefore introduce the temporal distance factor Ψ_{dk} which weights the contribution of the frame k. If c is the index of the current frame, we select $\Psi_{dk} = (1/2)^{|k-c|}$. Additionally, we can weight a frame based on the level of its additive noise. Neighborhoods from frames with a higher amount of noise can contribute less to a smooth solution and should thus have a lower weight. We obtain an estimate for the noise-level from the bandwidth s_k of Φ_{sk} for each frame k. We use these values to set the weighting factor $\Psi_{sk} = \exp(-s_k^2/\max_k\{s_k\}^2)$. By combining all elements, we determine the denoised point position as

$$\mathbf{p}_i' = \mathbf{p}_i - \frac{\sum_k \Psi_{dk}\Psi_{sk} \sum_{\mathbf{p}_j \in \mathcal{N}_k(\mathbf{p}_i)} \Phi_{dk}\Phi_{sk} \cdot [(\mathbf{p}_i - \mathbf{p}_j) \cdot \mathbf{n}_i] \cdot \mathbf{n}_i}{\sum_k \Psi_{dk}\Psi_{sk} \sum_{\mathbf{p}_j \in \mathcal{N}_k(\mathbf{p}_i)} \Phi_{dk}\Phi_{sk}}.$$

One advantage of our approach is that we do not necessarily need to compensate for motion between frames as the similarity of the whole temporal neighborhood is evaluated. If the motion is high, the similarity of the whole neighborhood will be low and it will only marginally contribute to the new point position. In this way, our approach also automatically accounts for scene changes.

4 Results

We demonstrate results of our denoising approach in Figures 4.1-4.3. We test our method on scanned data from various sources. We apply our algorithm to laser scanned models (Figs. 4.1+4.2) as well as to face and hand sequences which were acquired using a structured light scanner (Fig. 4.3). We compare our result with the bilateral filtering algorithm. Table 4.1 summarizes the timings for our results and the parameter settings used to generate them. All images are rendered using flat shading.

In Figure 4.1, we show the filtering efficiency of our approach on real-world laser scanned data. The images show that high-frequency noise on the Bimba model is removed after one iteration of our algorithm while lower-frequency details like hair, ear and eye are accurately preserved.

Figure 4.2 shows a comparison of bilateral filtering and our approach concerning feature preservation. Note that our algorithm creates a smoother result of the Turbine Blade model than bilateral filtering and preserves the sharp feature more accurately.

Figure 4.3 illustrates the results of the bilateral filter and our algorithm on three frames of the acquired structured light sequences. To filter the scans, we perform two iterations for each algorithm. In the first iteration, we filter with a larger kernel size to remove the stripe artifacts created due to the projection of regular line patterns onto the scanned object during the acquisition process. As the stripe pattern varies over time, our method filters across frames to increase the temporal stability of the smoothed sequence. We consider one frame before and after the current frame while filtering both sequences. High-frequency noise distributed over the whole scan does not show any temporal coherence. Therefore, we filter every frame separately with a smaller kernel size in the second iteration. We choose the parameters to optimize the result of each algorithm.

Figure 4.3 shows that our method removes the stripe artifacts and the high-frequency noise properly and achieves a more homogeneous result than the bilateral filter due to the comparison of geometric neighborhoods in-

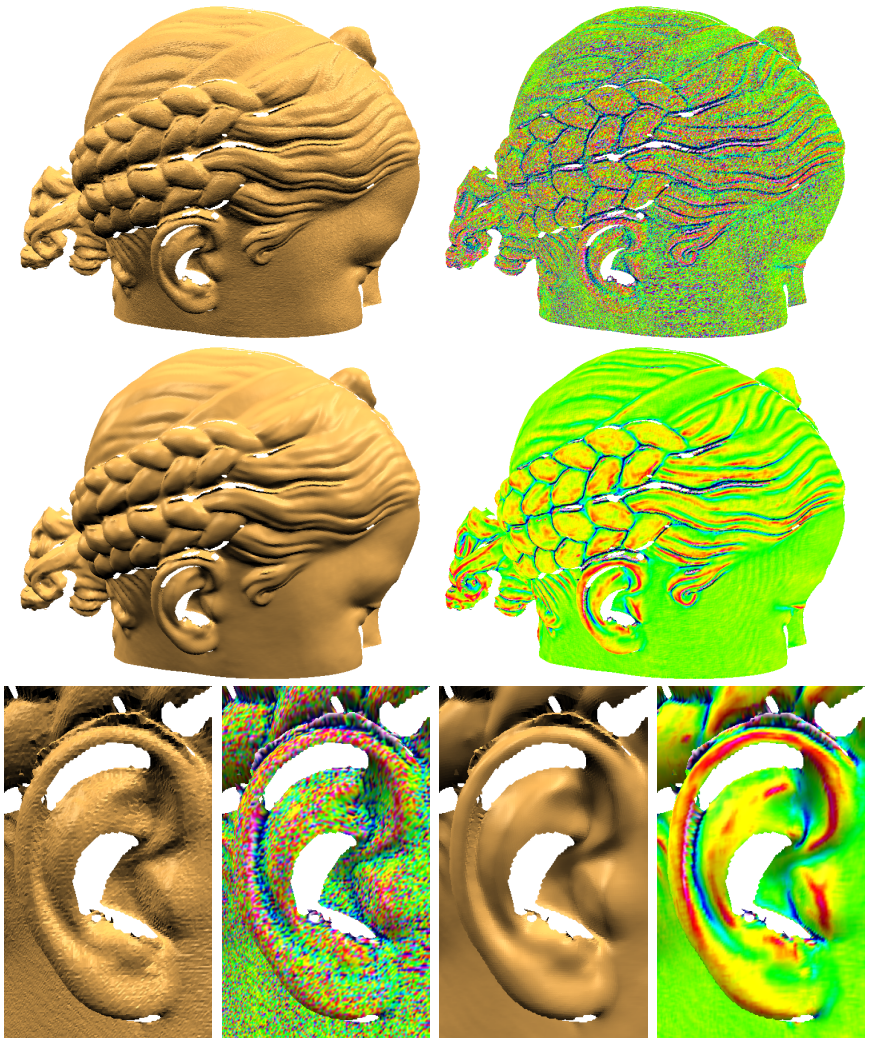

Figure 4.1: Results of our approach on a raw laser scan of the Bimba model (top row). The middle row shows the denoised result after one iteration of our algorithm. In the bottom row we show zooms of the noisy and denoised ear of the model and the corresponding mean curvature visualizations. Notice that high-frequency noise is nicely removed while details in hair, ear and eye regions are accurately preserved.

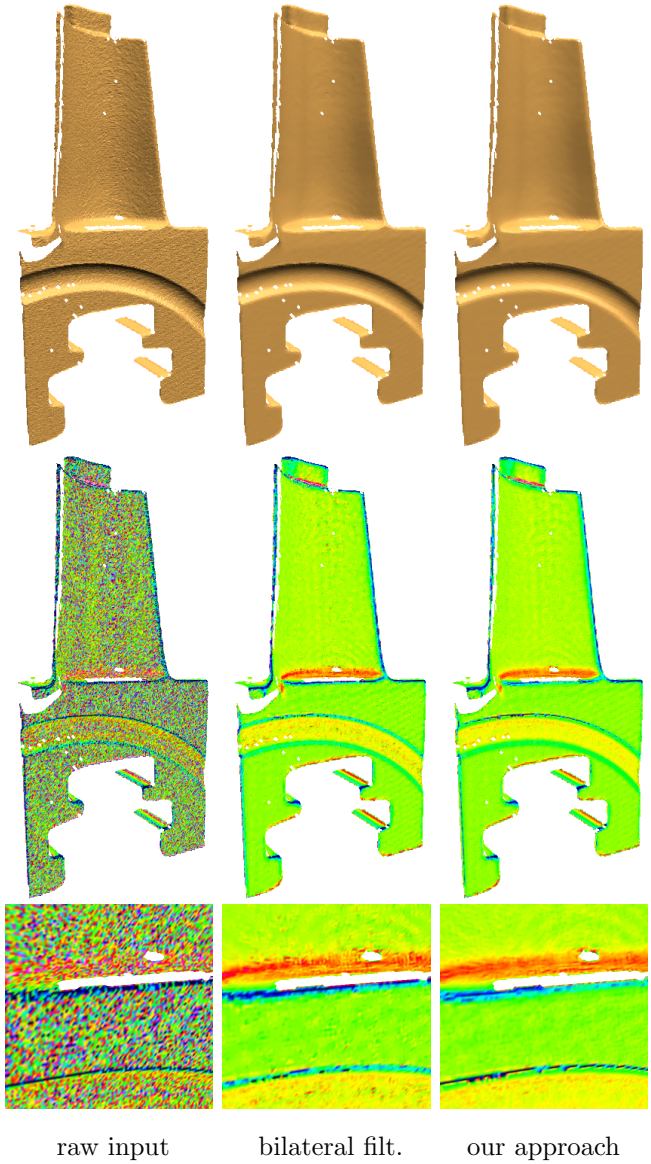

Figure 4.2: Comparison of feature-preservation properties of bilateral filtering and our approach on a laser range scan of the Turbine Blade model. The zoomed mean curvature visualizations show that our approach preserves sharp features more accurately than bilateral filtering while simultaneously producing a slightly smoother result.

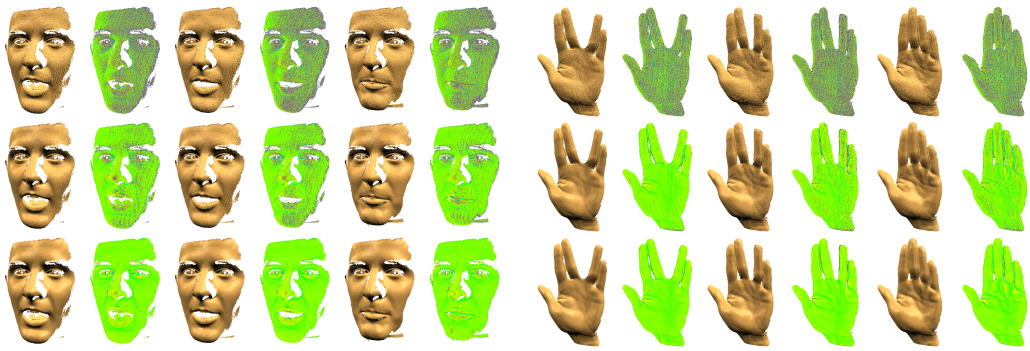

Figure 4.3: Denoising results for two acquired noisy range sequences. The raw input from the structured light scanner (top row) is denoised using bilateral filtering [9] (middle row) and our technique (bottom row). Coloring by mean curvature is used to illustrate the smoothness of the range data.

bilateral filtering

model	P	Sim	$\mathcal{N}(\cdot)$	time/frame
Blade	59K	–	11x11	2.81s
Face Sequence	192K	–	21x21	30s
(50 frames)		–	11x11	10s
Hand Sequence	131K	–	21x21	23s
(80 frames)		–	11x11	7s

our approach

model	P	Sim	$\mathcal{N}(\cdot)$	time/frame
Blade	59K	5x5	7x7	5.6s
Bimba	212K	5x5	7x7	7.4s
Face Sequence	192K	11x11	19x19	285s
(50 frames)		5x5	11x11	18s
Hand Sequence	131K	11x11	19x19	203s
(80 frames)		5x5	11x11	13s

Table 4.1: Parameter settings and timings for the results presented in this paper. The parameter P labels the average number of input points per frame. Sim denotes the size of the neighborhood considered to compute the similarity measure of our algorithm. All results were computed on a 2.66 GHz Pentium 4 with 1.5 GB of RAM.

stead of points. Since our algorithm exploits temporal coherence, we achieve stable filtering results along the time domain. Furthermore, our algorithm accurately preserves high-curvature regions, for instance, at the eyes and the lips of the face scan.

5 Conclusions & Future Work

We present a non-local neighborhood filtering technique for the accurate denoising of static and time-varying range data. To our knowledge, our approach is the first method which is designed to denoise time-varying geometric data. We show that it is able to preserve fine surface features, produces a better smoothing result than previous state-of-the-art neighborhood filters, and is easy to implement. In the future, we plan to enrich our algorithm by additional attributes like color which is usually acquired simultaneously with the geometric data. We believe that combining several attributes during the filtering process will further increase the performance of neighborhood filtering schemes.

Acknowledgements

The Bimba and Turbine Blade datasets are courtesy of the AIM@SHAPE shape repository. This work was supported in part by the European FP6 NoE grant 506766 (AIM@SHAPE).

Bibliography

[1] M. Alexa. Wiener filtering of meshes. In *Proceedings of Shape Modeling International*, pages 51–57, 2002.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, and C. T. Silva. Point set surfaces. *IEEE Visualization 2001*, pages 21–28, Oct. 2001.

[3] N. Amenta and Y. J. Kil. Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, August 2004. Proceedings of SIGGRAPH 2004.

[4] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM Transactions on Graphics*, 24(3):845–852, 2005. Proceedings of ACM SIGGRAPH 2005.

[5] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition (CVPR) 2005*, volume 2, pages 60–65, 2005.

[6] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH 99*, pages 317–324, 1999.

[7] T. K. Dey and J. Sun. Adaptive MLS surfaces for reconstruction with guarantees. In *Eurographics Symposium on Geometry Processing 2005*, pages 43–52, 2005.

[8] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics*, 24(3):544–552, 2005. Proceedings of ACM SIGGRAPH 2005.

[9] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3):950–953, 2003. Proceedings of ACM SIGGRAPH 2003.

[10] K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. *Computer Graphics Forum*, 23(3):391–400, 2004. Proceedings of EUROGRAPHICS 2004.

[11] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer. Bayesian point cloud reconstruction. *Computer Graphics Forum*, 2006. to appear.

[12] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3):943–949, July 2003. Proceedings of SIGGRAPH 2003.

[13] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.

[14] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.

[15] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *Signal Processing Letters*, 12(12):839–842, 2005.

[16] B. Mederos, L. Velho, and L. H. de Figueiredo. Robust smoothing of noisy point clouds. In *Proc. SIAM Conference on Geometric Design and Computing*, Seattle, USA, 2003. Nashboro Press.

[17] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *European Conference on Computer Vision (ECCV)*, 2006. to appear.

[18] M. Pauly and M. Gross. Spectral processing of point-sampled geometry. *Proceedings of SIGGRAPH 2001*, pages 379–386, 2001.

[19] M. Pauly, N. J. Mitra, and L. J. Guibas. Uncertainty and variability in point cloud surface data. In *Eurographics Symposium on Point-Based Graphics*, pages 77–84, Zurich, Switzerland, June 2004.

[20] O. Schall, A. G. Belyaev, and H.-P. Seidel. Robust filtering of noisy scattered point data. In M. Pauly and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics 2005*, pages 71–77, Stony Brook, New York, USA, June 2005.

[21] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of SIGGRAPH 95*, pages 351–358, 1995.

[22] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV)*, pages 839–846, 1998.

[23] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *Proceedings of Shape Modeling International*, 2006. to appear.

[24] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558, 2004. Proceedings of SIGGRAPH 2004.

[25] S. Zhang and P. Huang. High-resolution, real-time 3D shape acquisition. In *Computer Vision and Pattern Recongnition Workshop*, 2004.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: Index-access Optimized Top-k Query Processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems
MPI-I-2006-4-001	R. Mantiuk	?
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On Verifying Complex Properties using Symbolic Shape Analysis
MPI-I-2006-1-006	M. Kerber	Division-Free Computation of Subresultants Using Bezout Matrices
MPI-I-2006-1-005	I. Albrecht	?
MPI-I-2006-1-001	M. Dimitrios	?
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated Retraining Methods for Document Classification and their Parameter Tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An Emperical Model for Heterogeneous Translucent Objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric Calibration of High Dynamic Range Cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A.,. Magnor, H. Seidel	Joint Motion and Reflectance Capture for Creating Relightable 3D Videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and Design of Discrete Normals and Curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse Meshing of Uncertain and Noisy Surface Scattered Data
MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-004	Y. Kazakov	A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures
MPI-I-2005-2-003	H.d. Nivelle	Using Resolution as a Decision Procedure
MPI-I-2005-2-002	P. Maier, W. Charatonik, L. Georgieva	Bounded Model Checking of Pointer Programs
MPI-I-2005-2-001	J. Hoffmann, C. Gomes, B. Selman	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-008	C. Gotsman, K. Kaligosi, K. Mehlhorn, D. Michail, E. Pyrga	Cycle Bases of Graphs and Sampled Manifolds
MPI-I-2005-1-008	D. Michail	?
MPI-I-2005-1-007	I. Katriel, M. Kutz	A Faster Algorithm for Computing a Longest Common Increasing Subsequence
MPI-I-2005-1-003	S. Baswana, K. Telikepalli	Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs

MPI-I-2005-1-002	I. Katriel, M. Kutz, M. Skutella	Reachability Substitutes for Planar Digraphs
MPI-I-2005-1-001	D. Michail	Rank-Maximal through Maximum Weight Matchings
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H. de Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints
MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects