Available online at www.sciencedirect.com
ScienceDirect
Computer-Aided Design 40 (2008) 701–707
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

Adaptive feature-preserving non-local denoising of static and time-varying range data

Oliver Schall*, Alexander Belyaev, Hans-Peter Seidel

Computer Graphics Group, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany

Received 15 August 2007; accepted 26 January 2008

Abstract

We present a new method for noise removal on static and time-varying range data. Our approach predicts the restored position of a perturbed vertex using similar vertices in its neighborhood. It defines the required similarity measure in a new *non-local* fashion which compares regions of the surface instead of point pairs. This allows our algorithm to obtain a more accurate denoising result than previous state-of-the-art approaches and, at the same time, to better preserve fine features of the surface. Another interesting component of our method is that the neighborhood size is not constant over the surface but adapted close to the boundaries which improves the denoising performance in those regions of the dataset. Furthermore, our approach is easy to implement, effective, and flexibly applicable to different types of scanned data. We demonstrate this on several static and interesting new time-varying datasets obtained using laser and structured light scanners.

Keywords: Denoising; Region-based filtering; Non-local filtering; Feature-preservation; Anisotropy; Time-varying geometry

1. Introduction

With the increased usage of scanning devices, denoising of digitized models has become 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 image or shape. Especially, fine features are often lost if no special treatment is provided. The continuous progress achieved by a variety of denoising approaches which have been introduced in recent years in the fields of image processing and computer vision [1–3] as well as in computer graphics [4–8], clarifies that the development of reliable, accurate and versatile denoising techniques is a lively area of research and the foundation for a wide range of applications.

In this paper, we introduce a new method for range scan denoising called *non-local* neighborhood filtering inspired by a recent image processing technique [9] which presents remarkable results. The main idea of neighborhood filtering in general is to determine the denoised position of a vertex as a weighted average of similar vertices in its vicinity. In particular, the choice of the similarity measure has a strong influence on the efficiency of the denoising approach. Unlike previous neighborhood filters which determine the similarity of two points locally using only their positions and sometimes normals, our approach is *non-local* and defines the similarity by comparing regions of the surface around the vertices instead. This yields a more accurate denoising of the surface and improves the removal of higher noise levels compared to previous state-of-the-art filtering approaches. At the same time, fine shape features are better preserved. Another extension of our method is that the size of the regions which are used to determine the similarity measure and the restored point position are not constant over the whole dataset but adapted close to the boundaries which allows it to obtain a more accurate denoising result in these regions. In addition, our method is easy to implement and can be flexibly applied to different types of noisy data obtained using laser and structured light scanners. An example for the effectiveness of our method is presented in Fig. 1.

Our method addresses the denoising problem differently compared to most previous approaches since we denoise the range scans before they are combined within the scanning

* Corresponding author. Tel.: +49 6819325439.
E-mail addresses: schall@mpi-inf.mpg.de (O. Schall), belyaev@mpi-inf.mpg.de (A. Belyaev), hpseidel@mpi-inf.mpg.de (H.-P. Seidel).

0010-4485/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2008.01.011

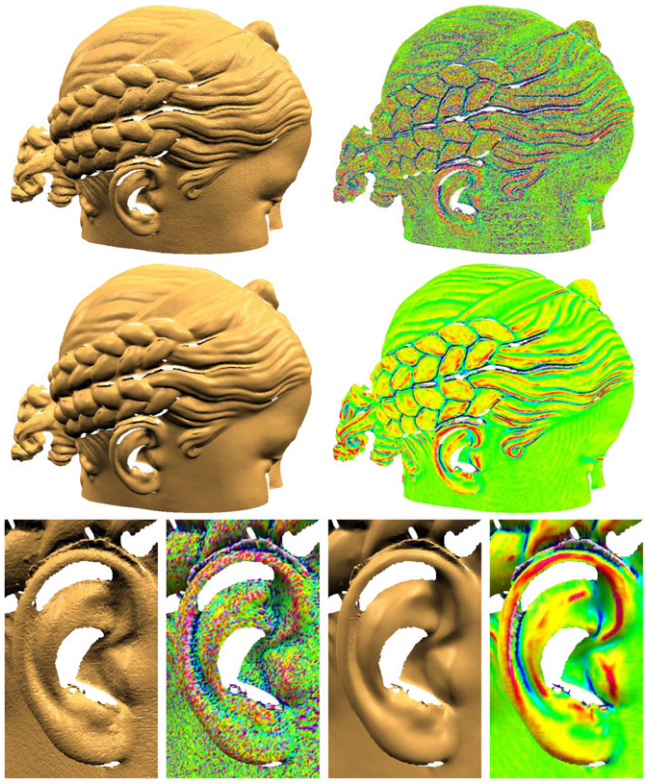

Fig. 1. A raw laser scan of the Bimba model (top row), the denoised result obtained using our algorithm (middle row), and a closer view of the noisy and denoised ear of the model (bottom row). For all images, the corresponding mean curvature visualizations are shown. Notice that high-frequency noise is properly removed after only one iteration of our method, while fine details in hair, ear and eye regions are accurately preserved.

pipeline. This is more efficient since the given structure of the data can be utilized in a simpler similarity measure which allows for a faster evaluation. Furthermore, our approach extends more naturally to dynamic range data whose acquisition has become feasible thanks to interesting improvements in scanning technology [10,11]. It is the first approach which is designed to filter time-varying geometric data, and we believe that proper denoising is the first step to open this new type of data up to various kinds of applications.

2. Related work

Denoising is a field of intensive research in image processing, computer vision, and computer graphics which comprises different classes of algorithms. The main approaches can be categorized into the following groups.

PDE approaches. Early works in image processing which introduced PDEs for denoising are anisotropic diffusion [1] and total variation minimization [2]. Desbrun et al. [5] proposed a geometric diffusion flow algorithm for irregular meshes and introduced the use of an implicit integration method which stabilizes the flow to allow larger time steps. Later, a new variant of anisotropic mean curvature flow which preserves nonlinear features was introduced [7]. This approach was extended in [12] in order to denoise point clouds.

Spectral techniques. Taubin [4] first introduced signal processing on meshes based on the definition of the Laplacian operator on surfaces. In [13], Wiener filtering was applied to meshes. Pauly and Gross [14] created a spectral decomposition of a point cloud and denoised it by manipulation of the spectral coefficients.

Neighborhood filtering. Neighborhood filters were early addressed in [15]. In a later work, Tomasi and Manduchi [3] introduced the well-known bilateral filter for images which was recently extended to non-local neighborhoods by Buades et al. [9]. Paris and Durand [16] proposed an interesting acceleration technique for the bilateral filter which allowed it to be mostly expressed as simple linear convolutions. Other works adapted neighborhood filters for video enhancement [17] and processing [18]. Further interesting research areas used the information contained in local neighborhoods for other tasks, such as texture synthesis [19,20] and mesh completion [21]. Fleishman et al. [8] proposed an anisotropic mesh denoising algorithm derived from the bilateral filter for images. Concurrently, Jones et al. [22] introduced a similar method based on robust statistics which uses local first-order predictors of a surface. Recent work of Yoshizawa et al. [23] extended the non-local image filter to meshes by computing a local RBF approximation to define the similarity measure. Our work approaches the problem in a different way by introducing a new similarity measure for range scans, which avoids the computation of a local approximation and thus allows a much faster evaluation. Additionally, our approach considers the denoising of time-varying geometric data.

Projection-based approaches. Algorithms that recently attracted the interest of many researchers are moving-least-squares (MLS) approaches. They were first proposed by Levin [24] and introduced to computer graphics by Alexa et al. [25]. 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. [26] applied the MLS projection for point cloud denoising. Amenta and Kil [27] 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. In a recent work, Fleishman et al. [28] chose a different approach and represented sharp features by defining piece-wise smooth moving least-squares surfaces using robust statistics. Dey and Sun [29] proposed the AMLS operator which provides reconstruction guarantees for the underlying surface of a point set with a non-uniform sampling density.

Statistical techniques. Pauly et al. [30] introduced a framework for measuring uncertainty in point-sampled geometry based on statistical data analysis which can be used, for instance, to merge range scans. Schall et al. [31] defined a global probability distribution function for a given noisy point set using locally defined kernels. Positions on a smooth surface were then found by moving every sample of the noisy dataset to maximum likelihood positions. Jenke et al. [32] proposed how to produce a smooth point cloud from a given noisy one

using Bayesian statistics. For image restoration, Awate and Whitaker [33] recently obtained remarkable results with an unsupervised, information-theoretic, adaptive filter which improved the predictability of a pixel by reducing the joint entropy between its neighborhood and other pixel neighborhoods.

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 [9] 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

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

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

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

Fig. 2 illustrates the computation of the region-based 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.

3.2. Static range data

We want to adapt this approach from the 2D plane to range data. This is not a straightforward task due to two fundamental differences between intensity images and range scans. Firstly, image pixels are usually aligned on a regular and equispaced grid which is in general not true for range images. Secondly, removing noise from a range scan is more complex than from intensity images since noise on a range scan is not necessarily additive to the surface but can, for instance, be dependent on the view of the scanner camera. In the case of images, noise is usually only additive to the intensity values. Therefore, we have

Fig. 2. 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 $\Phi(p, r)$.

to consider different displacement directions for vertices from range images which makes range scan denoising different and more difficult than image denoising.

We assume that the data is given in the form of points $\mathbf{p}_i$ which are arranged on a 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.

We find counterparts for the intensity values of an image by the distance of points $\mathbf{p}_j$ from a defined plane computed at a given point $\mathbf{p}_i$. Thereby, the orientation of the plane is orthogonal to the displacement direction $\mathbf{d}_i$ of a vertex $\mathbf{p}_i$. By determining the weighted average of the offsets, we find the displacement for $\mathbf{p}_i$, to remove the noise component from the range scan surface. In order to adapt to different types of noise such as view-dependent or additive noise, we choose different displacement directions $\mathbf{d}_i$. In the case of additive noise, we first estimate normals $\mathbf{n}_i$ for all points $\mathbf{p}_i$ by least-squares fitting to their local neighborhoods and choose $\mathbf{d}_i = \mathbf{n}_i$. For view-dependent noise, we select $\mathbf{d}_i$ to be the line-of-sight of the camera of the scanning device. In general, we determine the filtered points $\mathbf{p}'_i$ by computing

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

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 of our method from previous neighborhood filtering approaches for meshes 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:

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

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

This results in a more accurate filtering performance. We determine the point-wise difference of two square neighborhoods centered at $\mathbf{p}_i$ and $\mathbf{p}_j$ and project the distances onto the displacement direction $\mathbf{d}_i$. In the process, invalid points in the neighborhoods $\mathcal{N}(\mathbf{p}_i)$ and $\mathcal{N}(\mathbf{p}_j)$ are ignored. The result is the point-wise height difference of both neighborhoods which is averaged by the number of points inside the neighborhood (equal to the number of offsets $\|O\|$) to compute the non-local similarity $\mathrm{Sim}(\mathbf{p}_i, \mathbf{p}_j)$. Our similarity measure automatically compensates for translational movements of the neighborhoods along the defined plane. We also conducted experiments with a similarity measure that additionally compensates for rotations by aligning the directions $\mathbf{d}_i$ and $\mathbf{d}_j$ with each other before the similarity computation but it was more expensive to compute and did not improve the results significantly. 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 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 only 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 $\mathrm{Sim}(\mathbf{p}_i, \mathbf{p}_j)$ which regulates the homogeneity of the filtering result.

The size of the neighborhoods is usually chosen to be uniform over the dataset. According to our experiments the results of our approach can be improved if adaptive neighborhoods are used. If the neighborhood square $\mathcal{N}(\mathbf{p})$ of a point $\mathbf{p}$ contains insufficient information due to too many invalid points which is likely to happen mostly at the boundary of the dataset, our experiments showed that is it better to reduce the size of $\mathcal{N}(\mathbf{p})$ step-wise in order to optimize the ratio of its valid to its complete number of points. The effect of this enhancement is shown in Fig. 3. It is important to mention that the adaptation of the neighborhood size is optional and that our method does not require special boundary treatment which is important as scanned data is not closed and often has holes.

3.3. Time-varying range data

Building on the previous section, we extend our approach to process 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.

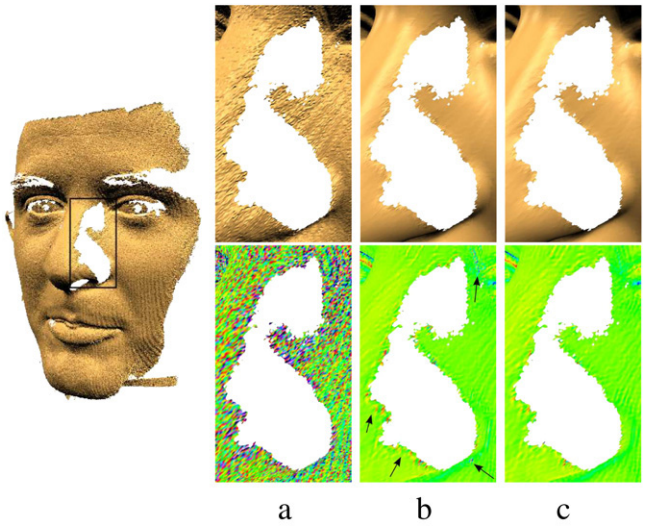

Fig. 3. A face scan obtained using a structured light scanner (a) is denoised using uniform neighborhood sizes (for $\mathcal{N}(\cdot)$ and Sim) (b) and using adaptive neighborhoods (c). Figure (c) shows that the denoising result close to the boundary is improved and that artifacts are removed.

Therefore, we extend $\mathcal{N}(\mathbf{p}_i)$ which is usually only defined as a spatial neighborhood by the temporal domain. This means that we choose sample points for $\mathcal{N}(\mathbf{p}_i)$ 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 points in different frames. We adapt the bandwidths of Φ_d and Φ_s depending on the frame with which $\mathbf{p}_j$ is associated. We detect the parameters automatically as described in Section 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 its noise level. 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

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

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.

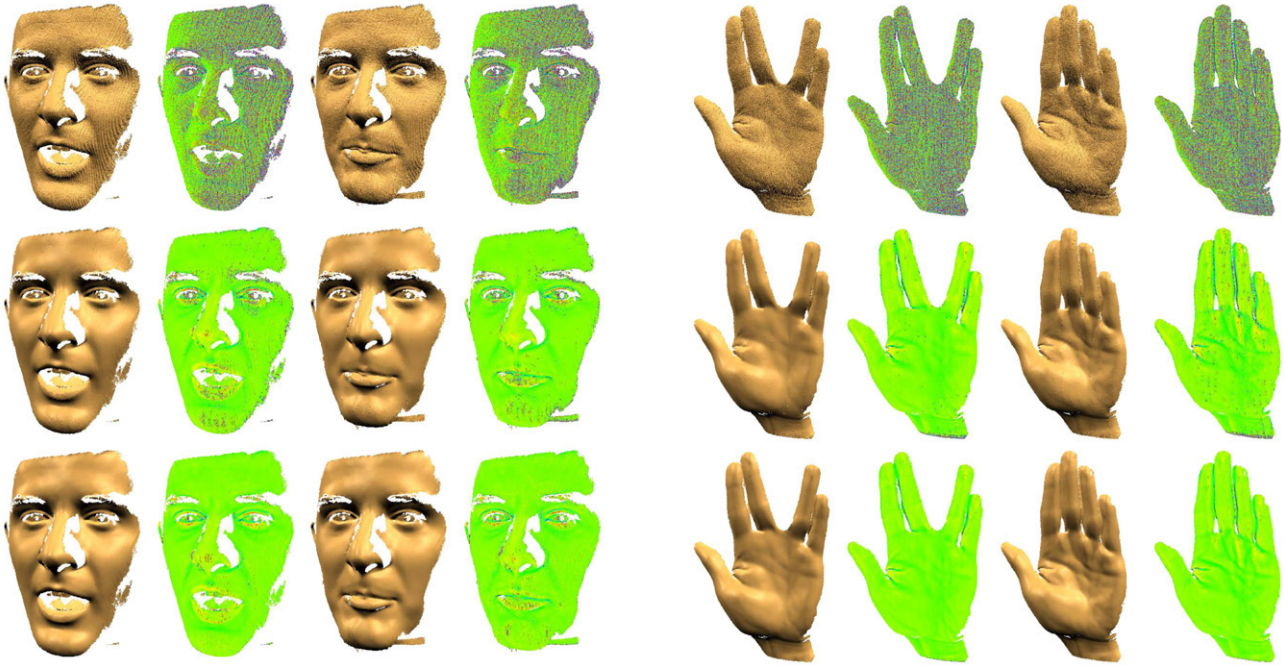

Fig. 4. Denoising results for two acquired noisy range sequences. The raw input from the structured light scanner (top row) is denoised using bilateral filtering (middle row) and our technique (bottom row). Coloring by mean curvature is used to illustrate the smoothness of the range data. Note that our algorithm is able to remove noise more properly while accurately preserving surface features.

Table 1
Parameter settings and timings for the results presented in this paper

Model	P (K)	Sim	$\mathcal{N}(\cdot)$	Time/frame (s)
Bilateral filtering				
Blade	59	–	11×11	2.8
Face sequence	192	–	23×23	94
(2 iterations)		–	11×11	16
Hand sequence	131	–	23×23	68
(2 iterations)		–	11×11	13
Our approach				
Blade	59	5×5	7×7	5.6
Bimba	212	5×5	7×7	7.4
Bust	358	9×9	13×13	240
(2 iterations)		5×5	11×11	48
Face sequence	192	9×9	13×13	95
(1 iteration)		5×5	11×11	18
Hand sequence	131	9×9	13×13	69
(1 iteration)		5×5	11×11	13

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.

In this way, our approach also automatically accounts for scene changes.

4. Results

We demonstrate results of our denoising approach in Figs. 1 and 4–6. We test our method on scanned data from various sources. It is applied to laser scanned models (Figs. 1–5) as well as to a face and hand sequence (Fig. 4) and to the Bust model (Fig. 6) which were acquired using two different structured light scanners. We compare our result to the bilateral filtering algorithm. Table 1 summarizes the timings for our results and the parameter settings used to generate them.

In Fig. 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 only one iteration of our algorithm while lower-frequency details such as hair, ear and eye are accurately preserved.

Fig. 5 shows a comparison of the bilateral filter and our approach concerning feature preservation. Note that our algorithm creates a smoother result of the Turbine Blade model than the bilateral filter while preserving sharp features more accurately.

Fig. 4 illustrates results of the bilateral filter and our algorithm on two frames of two acquired structured light sequences. To filter the scans, we apply two different kernel settings for each algorithm. First, 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, we let both methods filter 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 subsequently filter every frame separately with a smaller kernel size. The parameters are chosen such that each algorithm has its optimal performance. Our experiments show that using a larger kernel size for the bilateral filter than chosen in this comparison does not yield a smoother result. Instead, we allow the bilateral filtering algorithm to iterate twice over the sequences with both kernel sizes to make the runtimes of the bilateral filter and our algorithm match (see Table 1). Fig. 4 shows that our method

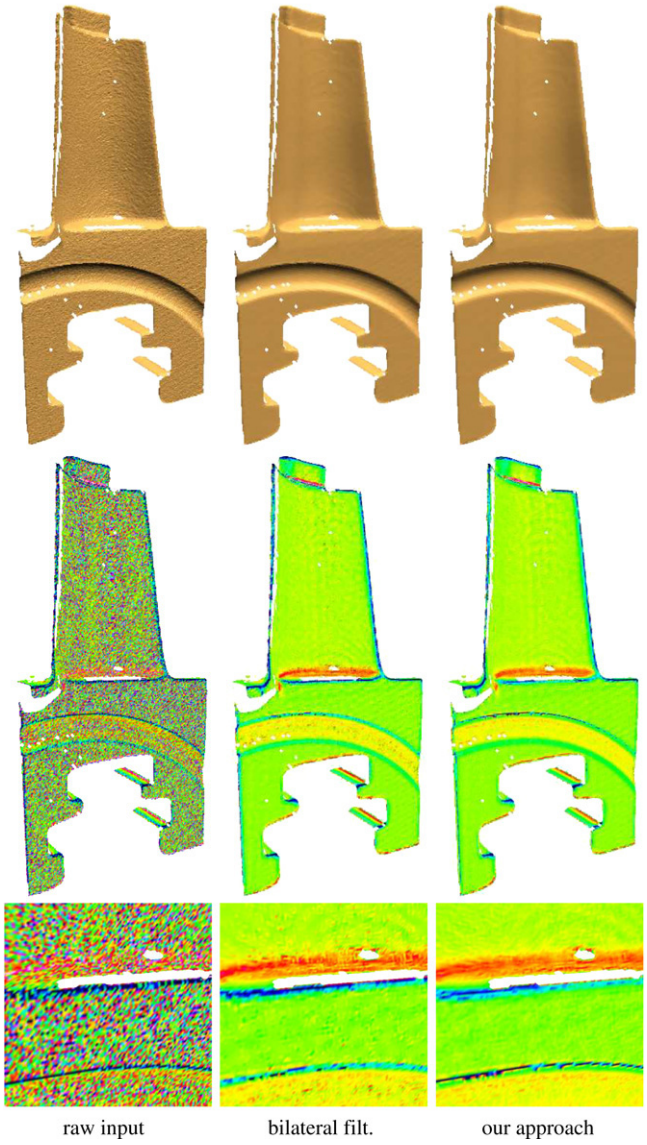

Fig. 5. Comparison of feature-preservation properties of bilateral filtering and our approach on a laser range scan of the Turbine Blade model. The zoomed mean curvature visualizations show that our approach preserves sharp features more accurately than bilateral filtering while simultaneously producing a smoother result. This can be observed on a more uniform mean curvature distribution after applying our algorithm compared to bilateral filtering. At the same time, our approach preserved the line feature present in the raw input more accurately.

removes the stripe artifacts and high-frequency noise properly and achieves a more accurate result than the bilateral filter due to our region-based definition of the similarity measure which adds more geometric information into the filtering process. Furthermore, our algorithm preserves high-curvature regions accurately, for instance, the eyes of the face scan and the wrinkles of the hand model.

In Fig. 6, we illustrate the adaptability of our method to different noise models. This is an important property since many previous approaches are bound to the case of additive noise where it is assumed that vertices are mostly displaced into the normal direction of the underlying smooth surface of the range scan. This yields a bad denoising performance if the noise

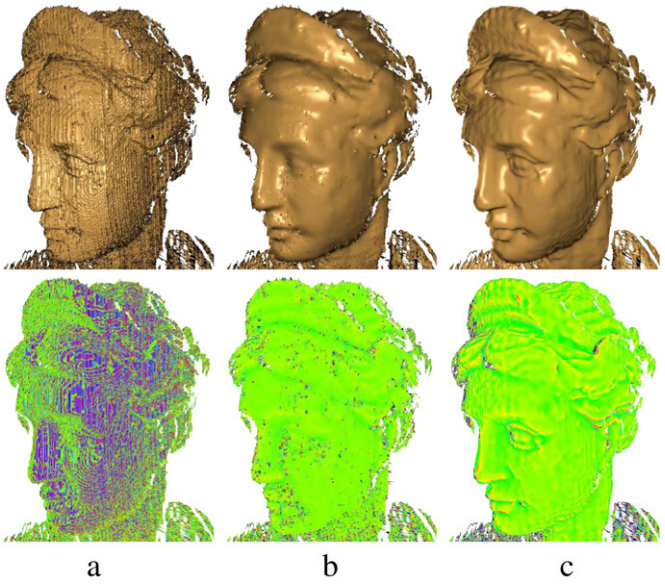

Fig. 6. The raw structured light scan of a Bust model (a) is corrupted by view-dependent noise. Simple denoising using an additive noise model, which is usually assumed or required by previous approaches, does not deliver satisfactory results (b). Due to the generality of our method, it is able to utilize additional information as, for instance, the camera viewing direction. This allows it to obtain a more accurate and detailed denoising result (c). For the computation of the results in (b)+(c) the same number of iterations and parameters are used for our method (see Table 1).

present in the data differs largely from the imposed model. Fig. 6(a) shows an example with an unprocessed structured light scan which is corrupted by view-dependent noise. By simply denoising the dataset with our method using the additive noise model, it is clearly noticeable that surface features are not well preserved and that smoothing artifacts are introduced (see Fig. 6(b)). In Fig. 6(c), we use the identical parameter settings for our denoising algorithm but use the viewing direction of the scanner, which is usually available from the scanner calibration data, for the displacement directions $\mathbf{d}_i$. This yields a significantly more accurate denoising result and a better preservation of high-curvature regions.

5. Conclusions and future work

In this paper, we presented a new similarity-based neighborhood filtering technique for static and dynamic range data which is the standard output of scanning devices and, in particular, of recently developed 3D video cameras. We introduced a new non-local similarity measure which determines the resemblance of two points on the surface not only by utilizing their local properties like position or normal but by also comparing the region of the surface surrounding the vertices instead. We demonstrated on several different types of scanned data that the idea of adding context information to the similarity definition allows our method to produce a more accurate denoising result than previous state-of-the-art approaches while having a better feature preservation. Additionally, we showed that the usage of adaptive neighborhoods improves the denoising result in the vicinity of boundaries of the given input. Furthermore, our method is easy to implement and flexibly adaptable to scans

with different noise properties. It thus delivers a practical, versatile and powerful tool for filtering range data. In this way, our approach naturally fits into the scanning pipeline by denoising range scans before they are combined for further processing, which is more efficient since the given structure of the data can be utilized. Furthermore, we showed an interesting extension of our approach for filtering time-varying geometric data which is important since we expect a wider use of 3D video cameras in the future. It exploits the temporal coherence of the sequence in order to guarantee smoothness along the time domain.

In the future, we plan to enrich our algorithm by additional attributes such as color which are 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

We would like to thank Sören König for providing us with the range scan of the Bust model. The Bimba and 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).

References

[1] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12(7):629–39.
[2] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D 1992;60:259–68.
[3] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision. 1998. p. 839–46.
[4] Taubin G. A signal processing approach to fair surface design. In: Proceedings of SIGGRAPH 95. 1995. p. 351–8.
[5] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH 99. 1999. p. 317–24..
[6] Tasdizen T, Whitaker R, Burchard P, Osher S. Geometric surface smoothing via anisotropic diffusion of normals. In: Proceedings of IEEE visualization 2002. Washington (DC, USA): IEEE Computer Society; 2002. p. 125–32.
[7] Hildebrandt K, Polthier K. Anisotropic filtering of non-linear surface features. Computer Graphics Forum 2004;23(3):391–400.
[8] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. ACM Transactions on Graphics 2003;22(3):950–3.
[9] Buades A, Coll B, Morel JM. A non-local algorithm for image denoising. Computer Vision and Pattern Recognition 2005 2005;2:60–5.
[10] Zhang L, Snavely N, Curless B, Seitz SM. Spacetime faces: High resolution capture for modeling and animation. ACM Transactions on Graphics 2004;23(3):548–58.
[11] Davis J, Nehab D, Ramamoothi R, Rusinkiewicz S. Spacetime stereo: A unifying framework for depth from triangulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2005;27(2):296–302.
[12] Lange C, Polthier K. Anisotropic fairing of point sets. Computer Aided Geometric Design 2005;22(7):680–92. [special issue].
[13] Alexa M. Wiener filtering of meshes. In: Proceedings of shape modeling international. 2002. p. 51–7.
[14] Pauly M, Gross M. Spectral processing of point-sampled geometry. In: Proceedings of SIGGRAPH 2001. 2001. p. 379–86.
[15] Yaroslavsky LP. Digital picture processing. An introduction. Berlin, (Heidelberg): Springer Verlag; 1985.
[16] Paris S, Durand F. A fast approximation of the bilateral filter using a signal processing approach. In: European conference on computer vision. 2006.
[17] Bennett EP, McMillan L. Video enhancement using per-pixel virtual exposures. ACM Transactions on Graphics 2005;24(3):845–52.
[18] Mahmoudi M, Sapiro G. Fast image and video denoising via nonlocal means of similar neighborhoods. Signal Processing Letters 2005;12(12): 839–42.
[19] Efros AA, Leung TK. Texture synthesis by non-parametric sampling. International Conference on Computer Vision, ICCV'99, vol. 2. 1999. p. 1033–8.
[20] Wei L-Y, Levoy M. Fast texture synthesis using tree-structured vector quantization. In: Proceedings of ACM SIGGRAPH 2000. New York (NY, USA): ACM Press/Addison-Wesley Publishing Co.; 2000. p. 479–88.
[21] Sharf A, Alexa M, Cohen-Or D. Context-based surface completion. ACM Transactions on Graphics 2004;23(3):878–87.
[22] Jones TR, Durand F, Desbrun M. Non-iterative feature-preserving mesh smoothing. ACM Transactions on Graphics 2003;22(3):943–9.
[23] Yoshizawa S, Belyaev A, Seidel H-P. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In: Proceedings of shape modeling international. 2006. p. 38–44.
[24] Levin D. The approximation power of moving least-squares. Mathematics of Computation 1998;67(224):1517–31.
[25] Alexa M, Behr J, Cohen-Or D, Fleishman S, Silva CT. Point set surfaces. IEEE Visualization 2001;(2001):21–28.
[26] Mederos B, Velho L, de Figueiredo LH. Robust smoothing of noisy point clouds. In: Proc. SIAM conference on geometric design and computing. Seattle (USA): Nashboro Press; 2003.
[27] Amenta N, Kil YJ. Defining point-set surfaces. ACM Transactions on Graphics 2004;23(3):264–70.
[28] Fleishman S, Cohen-Or D, Silva CT. Robust moving least-squares fitting with sharp features. ACM Transactions on Graphics 2005;24(3):544–52.
[29] Dey TK, Sun J. Adaptive MLS surfaces for reconstruction with guarantees. In: Eurographics symposium on geometry processing 2005. p. 43–52.
[30] Pauly M, Mitra NJ, Guibas LJ. Uncertainty and variability in point cloud surface data. In: Eurographics symposium on point-based graphics. 2004, p. 77–84.
[31] Schall O, Belyaev AG, Seidel H-P. Robust filtering of noisy scattered point data. In: Pauly M, Zwicker M, editors. Eurographics symposium on point-based graphics 2005, 2005. p. 71–77.
[32] Jenke P, Wand M, Bokeloh M, Schilling A, Straßer W. Bayesian point cloud reconstruction. Computer Graphics Forum; 25(3).
[33] Awate SP, Whitaker RT. Unsupervised, information-theoretic, adaptive image filtering with applications to image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence 2006;28(3):364–76.