Robust Non-Local Denoising of Colored Depth Data

Benjamin Huhle, Timo Schairer, Philipp Jenke, Wolfgang Straßer
University of Tübingen, WSI/GRIS
Sand 14, 72076 Tübingen, Germany
{huhle, schairer, jenke, strasser}@gris.uni-tuebingen.de

Abstract

We give a brief discussion of denoising algorithms for depth data and introduce a novel technique based on the NL-Means Filter. A unified approach is presented that removes outliers from depth data and accordingly achieves an unbiased smoothing result. This robust denoising algorithm takes intra-patch similarity and optional color information into account in order to handle strong discontinuities and to preserve fine detail structure in the data. We achieve fast computation times with a GPU-based implementation. Results using data from a time-of-flight camera system show a significant gain in visual quality.

1. Introduction

In recent years, cameras that measure real world distances based on the time-of-flight principle are becoming increasingly popular. A wide range of applications in scene acquisition, virtual reality and entertainment in general rely on data captured with such devices. Cameras that use an active modulated light source and sensors produced in standard CMOS technology (*e.g.*, from PMDTec, Canesta or the SwissRanger) are prospective low-cost sensors which could be deployed in standard applications, *i.e.*, mass markets. Measurements, however, performed with this kind of sensors are afflicted with noise and contain frequent outliers due to limitations of the sensor and the illumination situation. Therefore, appropriate postprocessing of the depth data is essential for robust and photo-realistic models. There are several motives for denoising the data in sensor space instead of considering 3D point clouds: For most sensors we have to deal with (additive) noise in viewing direction which is implicitly modeled if we consider the data as a depth map. Furthermore, the filtering is performed earlier in the pipeline and subsequent steps such as the registration of several frames can rely on the refined data. In contrast, for common surface reconstruction techniques in 3D, *e.g.*, multiple merged frames are necessary – otherwise, occlusions would impede sound solutions. In practice, we believe that a two-stage data enhancement is necessary, comprising an early data filtering in sensor space and a subsequent surface reconstruction computed on a registered model. We propose a denoising algorithm that detects outliers in a first stage and yields unbiased and feature-preserving smoothing results. Our approach is based on an image restoration technique, namely the recently presented non-local means (NL-means) filter [2] that makes use of self-similarity in the data.

In the first part of this paper (Section 2) we give a brief overview of different outlier detection and smoothing techniques. In the second part we introduce a novel unified denoising algorithm (Section 3) and show results (Section 4) using data from a system consisting of a time-of-flight and a color camera.

2. Previous Work

2.1. Outlier Detection

Depth data from different scanning devices may show very different noise characteristics. A common approach is to classify two different origins: Every measurement given by the sensor is affected by a certain level of inaccuracy. In some cases, however, the measurements can completely fail due to the illumination situation or because of problematic properties of the sampled surfaces (*e.g.*, strongly non-Lambertian reflectivity). These outliers need to be excluded from further processing, not only because they degrade the model quality, but since they could also lead to biased reconstructions, *e.g.*, smoothing results.

Weyrich et al. [25] presented three criteria for the detection of outliers in 3D point clouds and computes a combined outlier probability. The *plane fit criterion* measures the distance of a point to the surface model that is estimated by its neighborhood. The *miniball criterion* measures the distance of a point to the cluster built by its neighborhood. The observation that a valid point may be one of the k-nearest neighbors of an outlier, yet an outlier will most likely not be part of the k-nearest neighbors of the valid point, is called

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

the *nearest-neighbor reciprocity criterion*. Wand et al. [24] state that these criteria follow the same basic idea, namely to look at local correlations of points. Valid points, in contrast to outliers, are expected to lie on a smooth manifold, *i.e.*, to form a linearly correlated two-dimensional subspace. They apply the framework of *tensor voting* [17] to compute the *surfaceness* of a point which is an indicator of the probability that the point lies on a surface. Correspondingly, the *curveness* and *junctionness* that are also delivered by tensor voting determine whether it belongs to a curve or exhibits any affiliation to other points in space, respectively.

2.2. Smoothing Algorithms

Denoising in general is a highly ill-posed problem and demands for some means of regularization. The filtering of 2D data such as digital images has been studied intensively for several decades now and a variety of techniques have been developed. Standard linear filters such as the Gaussian are low-pass filters that smooth the data by averaging over neighboring pixels. Thereby, noise is removed but the data is blurred and features such as edges are destroyed. Therefore, such low-pass filters are appropriate if a completely smooth reconstruction is desired, but especially on depth data of general scenes these filters are not applicable. A popular fast non-linear filter is the median filter. Here, each pixel is replaced by the median of its local neighborhood. An important fact, especially when dealing with depth data is that no additional "smearing" effect occurs even at high jumps since every pixel is replaced by a measurement that exists in the neighborhood of the original data. However, the trade-off between a good smoothing result and disturbing artifacts is not optimal.

2.2.1 Neighborhood Filters

Yaroslavsky [26] introduced a filter that restores a pixel by an average of its neighbors weighted by their similarity. The *bilateral* [23] or *SUSAN* [22] filter uses the same concept, however, instead of using a spatial box function as done in the Yaroslavsky filter, the neighboring pixels are continuously weighted by their spatial Euclidian distance. Given a pixel v, the filtered pixel value v' at position $\mathbf{i}$ computes as

$$v'(\mathbf{i}) = \frac{1}{Z_\mathbf{i}} \sum_{\mathbf{j} \in \mathbf{I}} v(\mathbf{i})\, e^{\frac{1}{\sigma_s}\|\mathbf{i}-\mathbf{j}\|_2^2}\, e^{\frac{1}{\sigma_r}(v(\mathbf{i})-v(\mathbf{j}))^2}, \qquad (1)$$

where Z_i is the normalization constant such that

$$Z_\mathbf{i} = \sum_{\mathbf{j} \in \mathbf{I}} e^{\frac{1}{\sigma_s}\|\mathbf{i}-\mathbf{j}\|_2^2}\, e^{\frac{1}{\sigma_r}(v(\mathbf{i})-v(\mathbf{j}))^2} \qquad (2)$$

and summations are over the whole 2D index set $\mathbf{I}$. This results in a weight assigned to each neighboring pixel, that takes into account spatial distance as well as distance in the range domain, according to the user-specified parameters σ_s and σ_r. The weighting scheme renders the filter discontinuity-preserving, since the influence of the neighboring pixels is limited (see Fig. 1). A theoretical derivation of the bilateral filter is given in [6] and the relation to extended nonlinear diffusion methods is described in [1]. The bilateral filter is very popular in its original field but was also successfully applied to a wide variety of applications (see [18]). The underlying idea of the bilateral filter is easily extended to multi-modal data resulting in the joint- or cross-bilateral filter (*e.g.* [19]) where the filter weights are based on the similarity determined on a different data source or from the combination of several data sources, respectively. Kopf et al. [14] use this technique to enhance depth maps exploiting the dependencies of discontinuities in depth and color domain. Bilateral filtering was also applied to meshes [7] as well as to 3D point clouds [24].

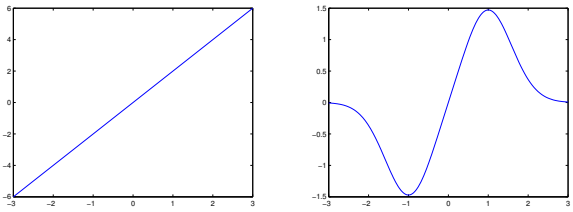

Figure 1. Influence functions (according to [10]) of Gaussian (left) and bilateral filter (right; using a Gaussian weight function).

2.2.2 Smoothing in an MRF-Framework

An overview and analysis of Markov Random Field (MRF) models for image analysis is given in [15]. The concept of an MRF is a probabilistic framework to infer hidden data values based on certain observations. A smoothness constraint, *i.e.*, the regularization is modeled by potential functions that represent statistical dependencies of neighbored pixels. The joint probability of the configuration of the graph can be maximized by minimizing the energy term of the corresponding Gibbs distribution according to the Hammersley-Clifford theorem. Using squared-distances in the potential functions, the minimization can be performed with standard techniques leading to a global minimum. This, however, yields a solution where discontinuities are blurred and fine details are lost. A variety of methods to overcome this effect in MRF models have been presented. *E.g.*, the *line process* model [8] switches off the regularization and introduces constant cost terms where discontinuities in the data occur, but demands for a sophisticated minimization method. Diebel and Thrun [4] use an MRF model to integrate high-resolution color images in order to enhance the lateral resolution of depth data. They introduced a weighting into the smoothness potentials applied

to upsampled depth data which reduces the regularization effect where discontinuities occur in the image data. The potential functions remain convex and hence, the minimization problem simple. This approach was also applied to image and depth data from combined time-of-flight and color camera systems [13, 11]. For image restoration, techniques have been presented recently, that make use of larger clique sizes and learn potential functions on a set of training data [20].

2.2.3 Non-Local Means Filter

The Non-Local Means (NL-Means) filter for image restoration was proposed recently by Buades et al. [2]. It follows a similar idea as the bilateral filtering approach, namely to restore a pixel by a weighted average of similar pixels

$$v'(\mathbf{i}) = \sum_{\mathbf{j} \in \mathbf{W_i}} w(\mathbf{i},\mathbf{j}) v(\mathbf{j}), \tag{3}$$

where $\mathbf{W_i}$ is a potentially large search window around $\mathbf{i}$. The similarity weight w, however, is determined in a different way. Instead of comparing the single pixel values at positions $\mathbf{i}$ and $\mathbf{j}$, patches (with index set $\mathbf{N}$) surrounding both pixels are taken into account. The weight is determined as

$$w(\mathbf{i},\mathbf{j}) = \frac{1}{Z_\mathbf{i}} e^{-\frac{1}{h}\sum_{\mathbf{k}\in\mathbf{N}} G_a(\|\mathbf{k}\|_2)(v(\mathbf{i}+\mathbf{k})-v(\mathbf{j}+\mathbf{k}))^2}, \tag{4}$$

with normalization constant

$$Z_\mathbf{i} = \sum_{\mathbf{j}\in\mathbf{I}} e^{-\frac{1}{h}\sum_{\mathbf{k}\in\mathbf{N}} G_a(\|\mathbf{k}\|_2)(v(\mathbf{i}+\mathbf{k})-v(\mathbf{j}+\mathbf{k}))^2}, \tag{5}$$

and the filtering parameter h. The pixelwise distances are weighted according to their offset $\mathbf{k}$ from the central pixel using a Gaussian kernel G_a with standard deviation a. In this way the self-similarity of the image is taken into account and even fine details that occur repeatedly can be distinguished from noise. A technique based on the same idea has also proven to be successful in the field of texture synthesis [5]. The NL-means filter was also used for simultaneous denoising and depth reconstruction from noisy image pairs [12]. Schall et al. [21] were the first to apply NL-means for denoising of 3D point data. They suppose that the data is given on a (not necessarily regular) 2D grid and compute the similarity weight according to the Euclidian distance in 3D. Depending on the type of noise, the point coordinates are corrected in the direction of the line-of-sight or along an estimated normal.

3. Robust Non-Local Denoising of Colored Depth Data

As mentioned earlier, depth data acquired with time-of-flight cameras contains a significant amount of erroneous measurements. Systematic errors can in part be corrected by means of calibration [16, 9], which reduces the noise level significantly, but the data remains noisy. In the following we present a novel technique to remove noise in depth data. Our algorithm explicitly handles outliers and preserves features and discontinuities in the data.

3.1. Outlier Detection

The NL-Means filter performs as an estimator of a pixel value given its surrounding. We employ this fact in order to detect abnormal values of central pixels in depth patches and classify these as outliers. When computing the similarity of patches, we have to rely on pixel values (in the destination as well as in the source patch) that are potential outliers and therefore use an iterative algorithm in the manner of expectation maximization (EM, [3]). We consider the probability distribution $p_{inlier}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right)$ of a pixel $v(\mathbf{i})$ given its surrounding $v(\mathbf{N_i^*})$, where $\mathbf{N_i^*} := \{\mathbf{i}+\mathbf{k}\}_{\mathbf{k}\in\mathbf{N}} \setminus \{\mathbf{i}\}$. The central pixel is not taken into account in order to obtain unbiased similarity estimates in case the pixel $\mathbf{i}$ is an outlier. The NL-Means in its original formulation (Eq. 3) computes the expectation value $E\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right) := \mu_\mathbf{i}$. Assuming that p_{inlier} follows a normal distribution, we compute the weighted variance

$$\sigma\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right) = \frac{1}{C_\mathbf{i}} \sum_{\mathbf{j}\in\mathbf{W_i}} w^*(\mathbf{i},\mathbf{j}) \left(v(\mathbf{j})-\mu_\mathbf{i}\right)^2, \tag{6}$$

normalized by

$$\frac{1}{C_\mathbf{i}} = \frac{\sum_{\mathbf{j}\in\mathbf{W_i}} w^*(\mathbf{i},\mathbf{j})}{\left(\sum_{\mathbf{j}\in\mathbf{W_i}} w^*(\mathbf{i},\mathbf{j})\right)^2 - \sum_{\mathbf{j}\in\mathbf{W_i}} w^*(\mathbf{i},\mathbf{j})^2}. \tag{7}$$

To handle outliers, the weights $w^*(\mathbf{i},\mathbf{j})$ denote the patch similarity (Eq. 4) multiplied by the product of the validity estimation of pixel $\mathbf{i}$ and $\mathbf{j}$. These probabilities are initialized with 1.0 and updated in successive iterations according to the mixture model

$$\begin{aligned} p_{mix}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right) \;=\; & \alpha\, p_{inlier}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right) + \\ & p_{outlier}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right), \end{aligned} \tag{8}$$

where $p_{outlier}$ follows a uniform distribution, *i.e.*, is constant and α is computed as

$$\alpha = \frac{1}{|\mathbf{W_i}|} \sum_{\mathbf{i}\in\mathbf{W_i}} p_{inlier}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right). \tag{9}$$

Summarizing, in each iteration we compute the mean and variance of the inlier distribution $p_{inlier} = \mathcal{N}(\mu_\mathbf{i}, \sigma_\mathbf{i})$ for each pixel $\mathbf{i}$, *i.e.*, one NL-Means step, as well as α and

$$p_{inlier}\left(v(\mathbf{i})|v(\mathbf{N_i^*})\right) = \theta\,\alpha \frac{1}{\sqrt{2\pi}\sigma_\mathbf{i}} e^{-\frac{(v(\mathbf{i})-\mu_\mathbf{i})}{2\,\sigma_\mathbf{i}^2}} \tag{10}$$

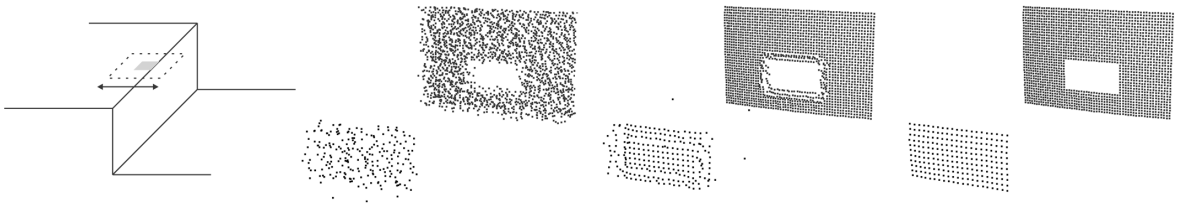

Figure 2. NL-Means Filter at a depth discontinuity: Smoothing in the depicted direction is prevented by a low similarity of the neighbored patches. Results on synthetic data (two distant surfaces at constant depth) from left to right: Input data with Gaussian noise, original NL-Means Filter, and our approach with additional weights ξ.

for the next iteration. A user-specified parameter θ is used to determine the sensitivity of the detector. Outliers are finally removed based on the binary decision $p_{inlier}\left(v(\mathbf{i})|v(\mathbf{N}_\mathbf{i}^*)\right) < 0.5$.

3.2. Smoothing

Once we have detected outliers in the depth map, the NL-Means filter can be used for smoothing. We exclude outlier pixels in the sum of Equation 3, *i.e.*, we sum over $\mathbf{j} \in \mathbf{W_i} \setminus \{\mathbf{k}\}$ where $v(\{\mathbf{k}\})$ have been classified as outlier values. Furthermore, we exclude outlier pixels when computing the patch similarity according to Equation 4. We experienced that, on depth data, the original NL-Means algorithm as it is used in the field of image restoration produces undesirable artifacts near strong discontinuities. The filter averages pixels along edges, but in a distance less than the neighborhood radius, smoothing will occur in neither direction perpendicular to the edge because of the strong dissimilarities of the patches. We illustrate this effect in Figure 2. In order to smooth over similar regions of depth also when a near edge is present, we introduce an additional term $\xi_{\mathbf{ij}}$ in the computation of the patch similarity (Eq. 4),

$$w(\mathbf{i},\mathbf{j}) = \frac{1}{Z_\mathbf{i}} e^{-\frac{1}{h}\sum_{\mathbf{k}\in\mathbf{N}} \xi_{\mathbf{ik}} G_a(\|\mathbf{k}\|_2)(v(\mathbf{i}+\mathbf{k})-v(\mathbf{j}+\mathbf{k}))^2}, \quad (11)$$

that results in an influence function similar to Fig. 1 for pixels in other patches of the search window. The new weighting factor

$$\xi_{\mathbf{ik}} = e^{-\frac{(v(\mathbf{i})-v(\mathbf{i}+\mathbf{k}))^2}{h^2}} \quad (12)$$

constrains the similarity comparison to regions of similar depths, using the same parameter h as in the computation of the inter-patch distances. This corresponds to the heuristic of searching self-similarities on the same surface and the weighting scheme leads to an effect similar to the adoption of NL-Means to 3D point data. The authors of [21] consider Euclidian distances in 3D instead of a Gaussian weighting in pixel coordinates (Eq. 4) and therefore minimize the influence of pixels that belong to very distant regions in depth. With our approach we try to explicitly distinguish different depth regions in order to achieve the same smoothing effect on distant surfaces where the sampling density is lower.

3.3. Integration of Color Information

Similar to [4, 14] who use additional color information for super-resolution, we consider it an additional cue for the estimation of $p\left(v(\mathbf{i})|\{v(\mathbf{i}+\mathbf{k})\}_{\mathbf{k}\in\mathbf{N}}\right)$ and compute a weight

$$w^{(u)}(\mathbf{i},\mathbf{j}) = \frac{1}{Z_\mathbf{i}^{(u)}} e^{-\frac{1}{h^{(u)}}\sum_{\mathbf{k}\in\mathbf{N}} \xi_{\mathbf{ik}}^{(u)} G_a(\|\mathbf{k}\|_2)\|\mathbf{u}(\mathbf{i}+\mathbf{k})-\mathbf{u}(\mathbf{j}+\mathbf{k})\|_2^2}. \quad (13)$$

corresponding to Equation 4. Here, $\mathbf{u}(\mathbf{i})$ denotes the color (we use RGB) assigned to pixel $\mathbf{i}$ and variables with superscript (u) are analogous to their counterparts in Equation 11. This yields the combined NL-Means formulation

$$v'_{uv}(\mathbf{i}) = \sum_{\mathbf{j}\in\mathbf{W_i}} w(\mathbf{i},\mathbf{j})\, w^{(u)}(\mathbf{i},\mathbf{j})\, v(\mathbf{j}), \quad (14)$$

resulting in an estimate of $v'(\mathbf{i})$ according to the distribution $p\left(v(\mathbf{i})|\{v(\mathbf{i}+\mathbf{k})\}_{\mathbf{k}\in\mathbf{N}}, \{\mathbf{u}(\mathbf{i}+\mathbf{k})\}_{\mathbf{k}\in\mathbf{N}}\right)$. The more reliable color data helps to preserve fine details in the depth map where discontinuities are present in both modalities.

4. Results

We apply the robust NL-Means Filter to data acquired with a *PMD [vision] 19k* time-of-flight camera. An additional industrial color camera is mounted ontop resulting in a 45mm parallax. Both cameras are calibrated laterally using a standard tool and a nearest-neighbor lookup of color values is performed, mapping the depth values into the color camera frame. No further depth calibration is applied.

Since the NL-Means is computationally expensive and straightforward to parallelize in a SIMD way, we outsource the algorithm to the GPU. Using the CUDA framework[1] we achieve computation times comparable to less demanding

[1] see the NVidia CUDA website: `http://www.nvidia.com/object/cuda_home.html`

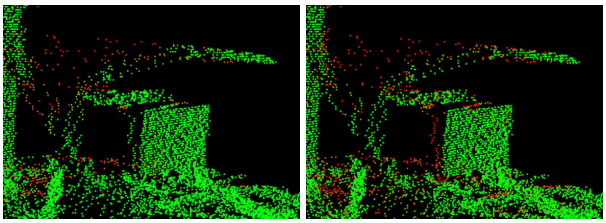

Figure 3. Outlier detection (detail view). Junctionness of the tensor-voting framework (left) and outlier estimate using our approach (right). Linear blending from red (=outlier) to green.

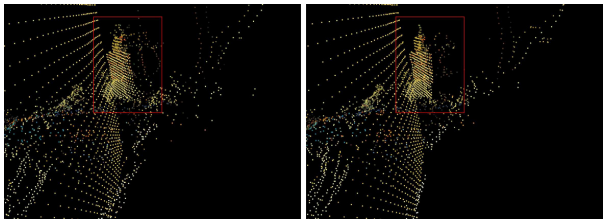

Figure 4. Effect of outlier removal on the non-local smoothing (detail view). Detection using tensor-voting (left) and our approach (right). Note the artifacts in the highlighted area.

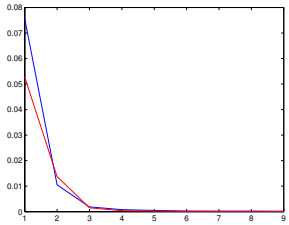

Figure 5. Average squared difference of the outlier probability per iteration for the depth maps of Fig. 6 (blue) and Fig. 9 (red).

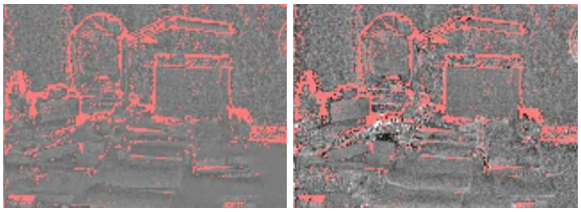

Figure 8. Method noise using the joint bilateral filter (left) and our robust non-local denoiser (right). Outliers are detected with the non-local denoiser and marked in red.

postprocessing methods. For the results, we use a patch size of $7x7$ (implying a Gaussian kernel with bandwidth $a = 1.5$) and a search window $\mathbf{W}$ of $17x17$ pixels. On a NVidia GeForce 8800GTX the algorithm takes 1.9 seconds performing 10 iterations in the outlier detection step.

A comparison of outlier detection techniques is not straight-forward, since the definition of "outlier" itself is rather vague and even for a single scene different persons may have different opinions which points to remove. We use the well-established tensor-voting framework as a reference. However, in contrast to Wand et al. [24] who use tensor-voting for surface reconstruction, we use the *junctionness* instead of the *surfaceness* value of the tensors to also conserve points lying on curves (see the desk light in Fig. 3). We compare the junctionness and the validity estimate p_{inlier} of our non-local outlier detector in Figure 3. The difference is only marginal and it is unclear which one is superior. However, our approach to prune outliers that would bias the smoothing result is justified by Figure 4 where we show the result of the whole denoising process applying our non-local smoothing step to both outlier results. The plot in Figure 5 illustrates the convergence of the outlier probability after only a few iterations.

Figures 6 and 7 show results using our robust non-local denoising filter on a challenging scene that contains fine structure as well as large differences in depth. For comparison, the same outlier set is also used for the MRF-based and the joint bilateral filter. The reference implementation of the joint bilateral filter uses the same approach as [14], yet working on the native resolution of the PMD camera. We observe a better handling of discontinuities and preservation of fine details using our appraoch. Especially, a better reconstruction is achieved for the desktop which is oriented almost parallel to the viewing direction. Using the (joint) bilateral filter, the influence of neighbored pixels at small discontinuities is approximately the same as in the simple Gaussian filter. Therefore, these regions tend to become oversmoothed.

According to [2] we also show the *method noise* of the non-local denoising algorithm in Fig. 8. Note that the interpretation is more difficult than in the case of image restoration. The uncertainties of the depth measurements are higher at object boundaries and therefore an enforced smoothing is not a-priori undesirable. The low smoothing effect of the joint bilateral filter is obvious in the area of the desktop.

Results from a second scene are shown in Figure 9.

5. Conclusion

We discussed several denoising algorithms for depth data that originate from the field of image restoration. Introducing an additional weighting scheme which takes into account color and intra-patch similarity, we presented a variant of the NL-Means Filter. Our filter explicitly handles outliers in the input data and is applicable to depth data with strong discontinuities. The unified denoising technique shows a performance in the outlier detection step comparable to the well-established tensor-voting framework that runs on 3D data. Compared to state-of-the-art tech-

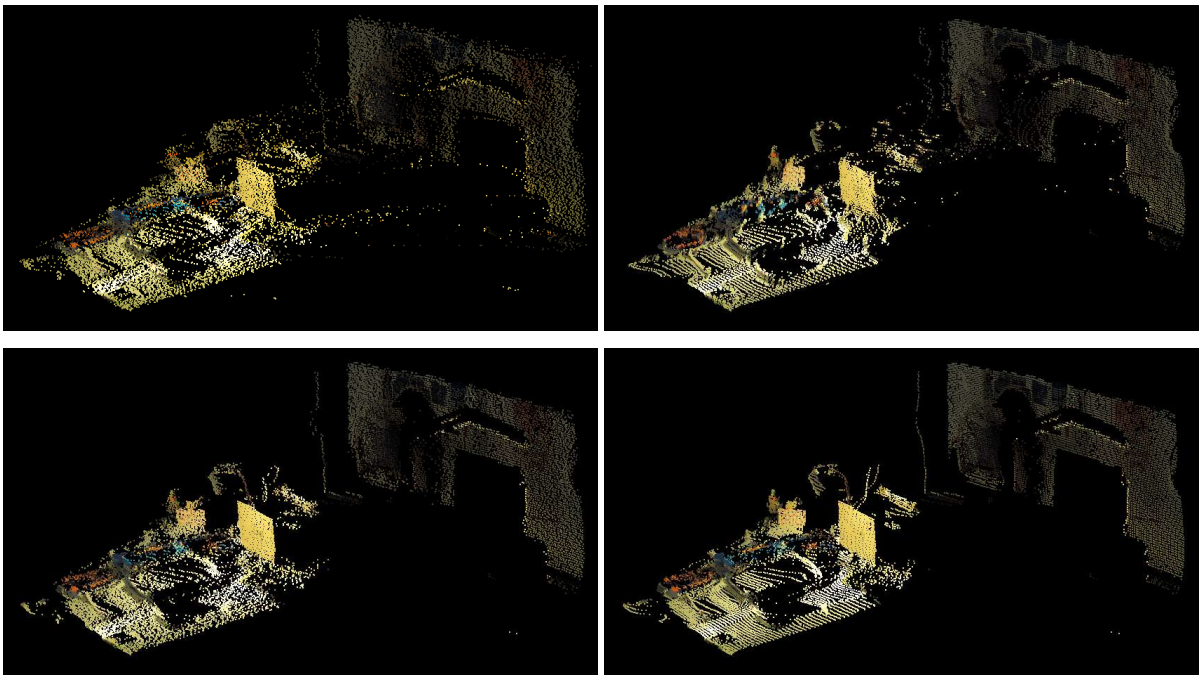

Figure 6. Denoising results. Raw data (top left), MRF-based denoising using squared-distance functions and color weights [4] (top right), joint bilateral filter (bottom left), robust non-local denoising (bottom right). Outliers (except top left) are removed using the iterative non-local approach prior to smoothing.

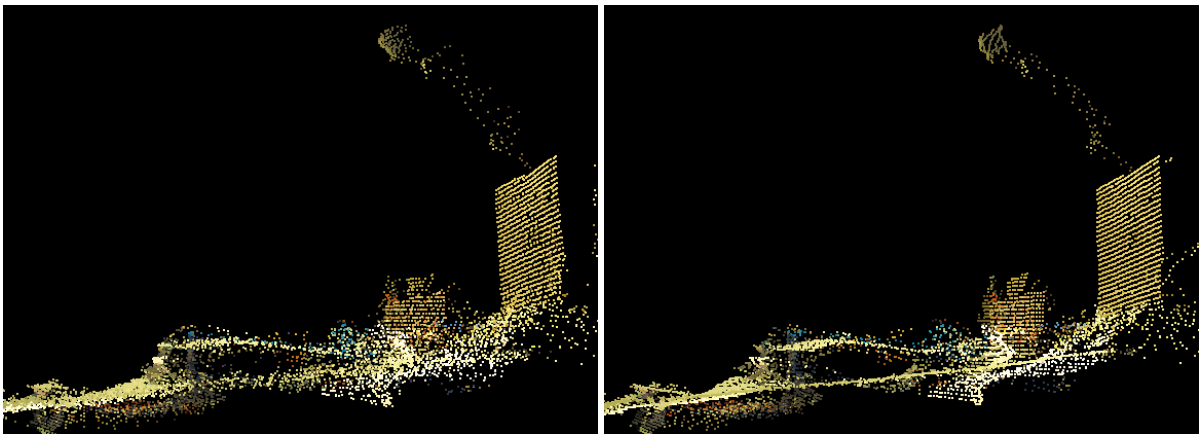

Figure 7. Detail view of the denoising result using joint bilateral filter (left) and our approach (right).

niques, the non-local denoising algorithm exhibits a superior preservation of fine detail and sharp edges while achieving smooth reconstructions where reasonable. The presented results using typical time-of-flight depth data with additional color information show a significant gain in visual quality.

References

[1] D. Barash and D. Comaniciu. A Common Framework for Nonlinear Diffusion, Adaptive Smoothing, Bilateral Filtering and Mean Shift. *Image and Video Computing*, 22(1):73–81, 2004.

[2] A. Buades, B. Coll, and J.-M. Morel. A Non-Local Algorithm for Image Denoising. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[4] J. Diebel and S. Thrun. An Application of Markov Random Fields to Range Sensing. In *19th Annual Conference on Neural Information Processing Systems (NIPS)*, 2005.

Figure 9. Result using our robust non-local denoising technique (right) and the input (left).

[5] A. A. Efros and T. K. Leung. Texture Synthesis by Non-parametric Sampling. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 1999.

[6] M. Elad. On the Origin of the Bilateral Filter and Ways to Improve It. *IEEE Transactions on Image Processing*, 11(10):1141–1151, 2002.

[7] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral Mesh Denoising. In *Proc. ACM SIGGRAPH*, 2003.

[8] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741, 1984.

[9] S. A. Gudmundsson, H. Aanæs, and R. Larsen. Environmental Effects on Measurement Uncertainties of Time-of-Flight Cameras. In *Proc. International Symposium on Signals Circuits and Systems (ISSCS)*, 2007.

[10] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, 1986.

[11] W. Hannemann, A. Linarth, B. Liu, G. Kokai, and O. Jesorsky. Increasing Depth Lateral Resolution Based on Sensor Fusion. *Int. J. on Intell. Systems and Techn. and App., Special Issue on Dynamic 3D Imaging*, 2008. (accepted for publication).

[12] Y. S. Heo, K. M. Lee, and S. U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using Non-local Pixel Distribution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] B. Huhle, S. Fleck, and A. Schilling. Integrating 3D Time-of-Flight Camera Data and High Resolution Images for 3DTV Applications. In *Proc. 3DTV CON '07*, 2007.

[14] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint Bilateral Upsampling. In *Proc. ACM SIGGRAPH*, page 96, New York, NY, USA, 2007.

[15] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Spinger, 2001.

[16] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD TOF-Camera. In *SPIE: Intelligent Robots and Computer Vision XXV*, volume 6764, pages 6764–35, 2007.

[17] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[18] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. A Gentle Introduction to Bilateral Filtering and its Applications. In *ACM SIGGRAPH Course*, 2007.

[19] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital Photography with Flash and No-Flash Image Pairs. In *Proc. ACM SIGGRAPH*, 2004.

[20] S. Roth and M. J. Black. Fields of Experts: A Framework for Learning Image Priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[21] O. Schall, A. Belyaev, and H.-P. Seidel. Feature-preserving Non-Local Denoising of Static and Time-Varying Range Data. In *Proc. ACM Symposium on Solid and Physical Modeling*, 2007.

[22] S. M. Smith and J. M. Brady. SUSAN – A new Approach to Low Level Image Processing. Technical report, Oxford University, 1995.

[23] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 839–846, 1998.

[24] M. Wand, A. Berner, M. Bokeloh, P. Jenke, A. Fleck, M. Hoffmann, B. Maier, D. Stanecker, A. Schilling, and H.-P. Seidel. Processing and Interactive Editing of Huge Point Clouds from 3D Scanners. *Computers & Graphics*, :, 2008. to appear.

[25] T. Weyrich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella, and M. Gross. Post-Processing of Scanned 3D Surface Data. In *Proc. Eurographics Symposium on Point-Based Graphics*, 2004.

[26] L. P. Yaroslavsky. *Digital Picture Processing. An Introduction*. Springer Verlag, 1985.