# Temporal Filtering of Depth Images using Optical Flow

Razmik Avetisyan Christian Rosenke Martin Luboschik Oliver Staadt

Visual Computing Lab, Institute for Computer Science
University of Rostock
18059 Rostock, Germany

{razmik.avetisyan2, christian.rosenke, martin.luboschik, oliver.staadt}@uni-rostock.de

## ABSTRACT

We present a novel depth image enhancement approach for RGB-D cameras such as the Kinect. Our approach employs optical flow of color images for refining the quality of corresponding depth images. We track every depth pixel over a sequence of frames in the temporal domain and use valid depth values of the same point for recovering missing and inaccurate information. We conduct experiments on different test datasets and present visually appealing results. Our method significantly reduces the temporal noise level and the flickering artifacts.

## Keywords

Temporal Filtering, Optical Flow, Depth Image Enhancement, RGB-D Sensor

## 1 INTRODUCTION

Today, commodity RGB-D cameras such as the Microsoft Kinect are very popular because of their affordability and the capability to output color and depth images at a high frame rate. They are widely used in computer graphics and virtual reality applications as a low-cost acquisition device.

However, while the color images contain fine details of the scene, the depth images have lower spatial resolution and suffer from extensive noise. The disturbance has a strong temporal component and is perceived as an annoying flickering, even if camera and scene are static. Depth images also contain holes where no depth measurements are available. See Figure 1 to get an impression of the artifacts. Before the depth data can be used in an application, it usually has to be enhanced. There are several existing approaches for this problem that are mostly based on spatial filtering [Chen et al., 2012, Camplani and Salgado, 2012a, Camplani and Salgado, 2012b, Garcia et al., 2013, Yang et al., 2013]. But due to the flickering nature of depth values those approaches oftentimes do not offer satisfactory results.

There are only very few methods that consider the temporal aspect of noise [Matyunin et al., 2011, Islam et al., 2015, Kim et al., 2010]. One reason for this may be the trouble with blurry object boundaries and ghosting artifacts introduced by temporal filtering of dynamic scenes. This happens as temporal filters usually combine depth values of the same pixel from different frames. When the part of the scene represented by that pixel changes over time, which is quite probable in a dynamic scene, then mixing the corresponding depth values is not valid and leads to the mentioned artifacts.

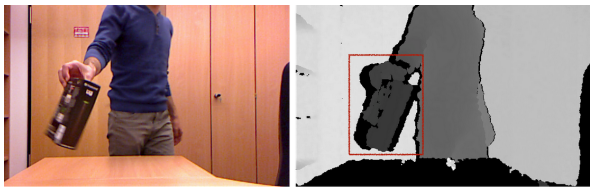

Figure 1: Color and depth images captured using a Kinect RGB-D camera. The depth image contains noise and flickering artifacts while the color image is more robust.

In this work we solve the aforementioned challenges and present a new temporal filtering approach for depth images. We propose to track the movement of objects in the depth image to consistently apply the temporal filter on the same parts of the scene, even if it moves. Based on the detected pixel movements, our method is able to enhance the image quality with standard filtering techniques applied to the temporal domain. However, tracking movements in depth image sequences is a very complicated problem which is even more hampered by their unstable nature, as mentioned above. To circumvent this difficulty, we present a new method for the tracking of movements. As RGB-D cameras simul-

taneously provide color and depth image, we decided to estimate the optical flow of consecutive color images in order to transfer the result to the corresponding depth images. Our idea benefits from the fact that color and depth cameras are usually located close to each other, that is, on the same baseline and with a very small offset. Therefore, we may safely assume that the motion of the imaged scene is induced almost equally on both cameras which eases the transfer.

Having a sound estimation of movement for all consecutive depth frames, we are able to trace back a certain displacement history for each pixel. This provides a one dimensional filtering field for every pixel, which can be processed with any standard 1D filter kernel, such as a simple Gaussian filter. We show that this way largely replaces inaccurate or noisy depth values by valid and stabilized ones. The method can be easily combined with other refinement strategies such as hole filling approaches. We validate our enhancement strategy using two publicly available test datasets and present visually appealing results.

## 2 RELATED WORK

There are a number of existing approaches that cope with the noise in depth images. Most of them represent classical spatial filtering methods. However, our work is not the first one proposing a temporal approach. A very relevant technique with respect to this article is presented in [Matyunin et al., 2011]. They propose a motion compensation strategy, but only for temporally smoothing depth images. The missing depth pixels are still being recovered from the neighboring pixels, and not from the temporally successive pixels. Furthermore, their approach is an offline approach. An online temporal approach is given in [Islam et al., 2015]. The authors propose to consider the history of depth pixels in the time domain but they do not track the movements. They use a simplified but well parallelizable least median of squares filter to robustly stabilize the depth values. Although their method performs well for static parts of the scene, it exhibits a lot of ghosting artifacts in dynamic parts. In [Kim et al., 2010] the authors propose a combined spatial and temporal depth enhancement method which even applies motion flow between successive color images to infer information about object motion in the corresponding depth images. However, they basically ignore this data in the dynamic parts of the depth images as they use it only to detect stationary parts. Based on this, they apply a bilateral filter to improve the quality which naturally fails in dynamic parts. [Hui and Ngan, 2014] enhance depth images captured from a moving RGB-D system. They also estimate the optical flow of consecutive color images. However, instead of building a temporal filter on top of the obtained data, their method estimates additional depth cues from the flow which are then combined with the original depth images. Their method is intended for mobile setups and cannot be applied to stationary cameras as mainly considered in our case.

Apart from that there are many standard spatial filtering approaches. However, some of them incorporate the information from the color image into the filtering, which relates them to our work. One, proposed in [Camplani and Salgado, 2012a, Camplani and Salgado, 2012b], uses a joint bilateral filter which combines depth and color information. It is working well for static scenes only. The work presented in [Chen et al., 2012] also uses a joint bilateral filter to fill the holes in the depth images. The corresponding color images are used to find and remove wrong depth values near to the edges. Their approach fails to work well for parts where the color image contains a dark region. Other works that incorporate color information for enhancing the quality of the corresponding depth images are presented in [Garcia et al., 2013, Yang et al., 2013]. These approaches provide quite good results in real-time. To sum up, the above mentioned approaches are reducing the noise by using spatial filters mostly. But overall there are few temporal filtering methods that remove the noise caused by moving objects while having stationary cameras.

## 3 PROPOSED METHOD

To fix the unstable nature of depth pixels captured by RGB-D cameras, we propose a new strategy that enables temporal filtering by keeping track of depth pixels in the time domain. We save a movement history for each depth pixel among a sequence of consecutive frames which is used to validate and correct pixels values. For tracking depth pixels in the time domain, our method employs optical flow [Radford and Burton, 1978], which describes the probable motion of pixels in pairs of consecutive depth frames of a video stream. As the depth stream is too noisy for the accurate estimation of optical flow, we calculate optical flow for the much more stable color video, usually delivered alongside depth data, and apply it for the depth pixels.

In our framework, we assume that an RGB-D sensor continuously provides a sequence of color and depth frame pairs $(I_i, D_i)$. By $I_i(x,y)$ we denote the color of pixel $(x,y)$ in the $i$-th color frame. Similarly, $D_i(x,y)$ refers to the depth value of pixel $(x,y)$ in the $i$-th depth frame. While receiving this data in real time, our method always keeps the latest $n$ image pairs. For every frame $(I_p, D_p)$ presently delivered, we use the information in the whole subsequence to produce an improved version $D'_{p-m}$ of the depth image $D_{p-m}$ in the sequence. Hence, every output frame is build on an $m$-element

preview and an $(n-m-1)$-element history. Clearly, the value of $n$ basically affects memory consumption whereas the value of $m$ influences the latency of our method.

Each incoming pair $(I_p, D_p)$ of frames is firstly inserted at the beginning of our monitored sequence while the oldest one, $(I_{p-n}, D_{p-n})$, is discarded. Next, we establish two motion fields $M_p, N_{p-1}$ between the new color frame $I_p$ and the previously first color frame $I_p - 1$. While $N_{p-1}$ describes the forward, that is, natural motion of pixels in time, $M_p$ helps to trace back movements. In $N_{p-1}$, each pixel $(x,y)$ holds a 2D vector $(u,v)$ describing the path taken by the pixel $(x,y)$ from $I_p - 1$ to $I_p$. More precisely, $N_{p-1}(x,y) = (u,v)$ states that the color value of pixel $(x,y)$ in the image $I_p - 1$ can be traced back to the pixel $(x+u, y+v)$ in the image $I_p$, that is,

$$I_{p-1}(x,y) \approx I_p(x+u, y+v). \tag{1}$$

While this makes it possible to follow the movement of a pixel along the sequence of frames, it does not help very much to tell where a pixel came from. Hence, here we use $M_p$ where $M_p(x,y) = (u',v')$ states that the color value at $I_p(x,y)$ can be traced back to the previous frame at $I_{p-1}(x+u', y+v')$. As we perform this procedure in every step, we can assume that we have motion fields $M_i$ and $N_{i-1}$ for the pair $I_i$ and $I_{i-1}$ of consecutive color frames for all $i$ in $\{p, ..., p-n+1\}$.

In the next step, we apply the estimated motion fields of the color image sequence to track the history and follow the future of pixels in the corresponding depth images. In particular, for every depth pixel $(x,y)$ in $D_{p-m}$, we traverse through the available $n$ depth frames following the respective motion vectors. That means, we obtain a sequence $(x_p, y_p), (x_{p-1}, y_{p-1}), \ldots, (x_{p-n+1}, y_{p-n+1})$ of pixel coordinates by setting

$$(x_i, y_i), = \begin{cases} (x,y), & \text{if } i = p-m, \\ (x_{i-1}, y_{i-1}) + & \\ \quad N_{i-1}(x_{i-1}, y_{i-1}), & \text{if } p \le i < p-m, \\ (x_{i+1}, y_{i+1}) + & \\ \quad M_{i+1}(x_{i+1}, y_{i+1}), & \text{if } p-m < i < n. \end{cases}$$

Ideally, this sequence accurately describes the past and prospective motion of the scene object represented at the pixel $(x,y)$ in frame $D_{p-m}$. That means, we can represent the depth of this object in another sequence $d_p, d_{p-1}, ..., d_{p-n+1}$ of $m$ prospective and $(n-m-1)$ historic depth values by defining $d_{p-i} = D_{p-i}(x_{p-i}, y_{p-i})$ for all $i \in \{0, \ldots, n-1\}$. Figure 2 illustrates this concept.

Recall that the motion fields are derived from the color image. That means for the identified depth sequence that we might get slightly varying depth values due to the $z$-movement of objects and because of the present noise.

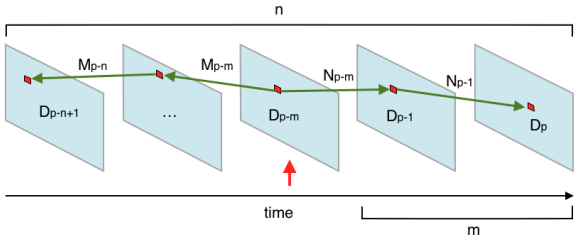


Figure 2: Motion compensated sequence of depth values. For each pixel $(x,y)$, we iterate over a short sequence of prospective and historic depth frames using the motion fields $M$ and $N$.

Finally, to stabilize the noise in depth image $D_{p-m}$, we basically filter the $n$ depth values of every pixel, which represents a temporal filtering approach. In our method we apply a weighted filter as follows:

$$D_{p-m}(x,y) = \frac{\sum_{i=0}^{n-1} \omega_i d_{p-i}}{\sum_{i=0}^{n-1} \omega_i} \tag{2}$$

The weights $\omega_i$ can be chosen to model certain filter kernels, as for instance a Gaussian filter:

$$\omega_i = e^{-(m-i)^2} \tag{3}$$

Beside static kernels like this, we also support motion dependent kernels, where the weight $\omega_i$ is determined by the amount of motion in frame $D_{p-i}$ at pixel $(x,y)$, that is, by the length of the vector $M_{d-i}(x,y)$, respectively of $N_{d-i-1}(x,y)$. This can be used to adaptively reduce the impact of highly dynamic depth frames in which a misinterpretation of real movements is more likely.

## 4 EXPERIMENTS

For experiments, we fixed the parameters of the method described in Section 3. To keep the latency of our approach low and minimize the ghosting artifacts, we chose to consider a 5-frame history by letting $n = 5$ and we set $m = 0$. This means that there was no preview. Furthermore, to keep the setup simple and to demonstrate our method's potential, we decided to just use a plain averaging filter in the 1D temporal domain. Hence, we set $\omega_i = 1$ for all $i$. The optical flow was estimated in real time by the method of [Brox et al., 2004] implemented in hardware.

To test the performance of our approach, we have applied different datasets captured with a Kinect camera, each containing at least one moving subject or object. Beside some self-created datasets, we conducted experiments on two publicly available datasets from [Camplani and Salgado, 2014]. Figure 3 demonstrates the method's visually appealing results using our own test sets. Apparently, as seen in the right image, noise and missing depth information, that essentially disturb

the original depth frame in the left image are noticeably fixed or at least reduced by our approach. We like to point out, that the visual improvement covers both, static and dynamic parts of the scene. Furthermore, we also significantly remove flickering, that is, temporal artifacts.

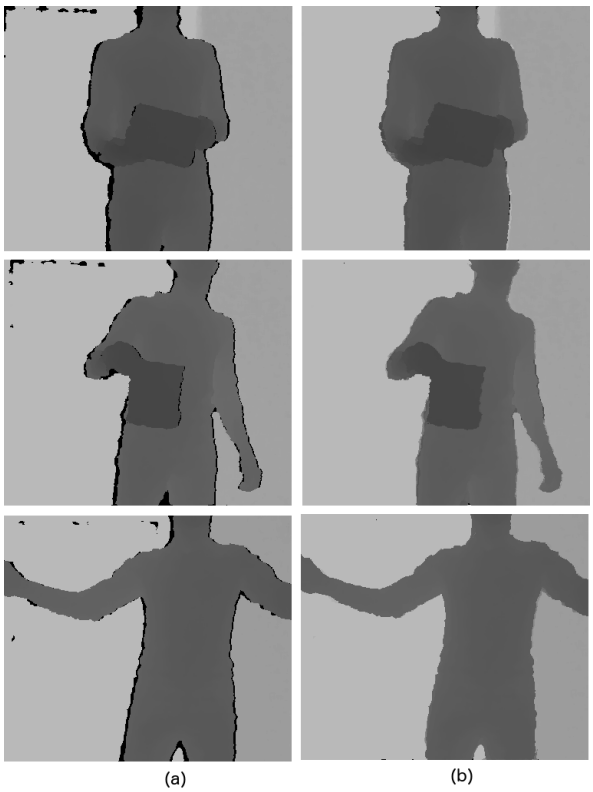


Figure 3: Results for our own datasets. (a) original raw depth images. (b) depth images enhanced by our approach.

Using the datasets from [Camplani and Salgado, 2014], we can also compare the performance of our approach to another state of the art spatial filtering technique for depth image enhancement as described in [Garcia et al., 2013]). For both sets, as depicted in Figure 4, our method fixes most of the missing information and reduces temporal noise for both, static and dynamic parts. Beside that, we get nicer and finer edges around objects.

Limitations of our approach are twofold. Firstly, missing data or noise that stays persistently in one region of the depth image sequence can not be recovered by our temporal filtering approach. In this case, spatial filters, as presented in [Garcia et al., 2013], may perform better. Secondly, artifacts introduced by the motion fields can essentially influence the quality of the output. Even though the color images are more stable and of a higher resolution, it happens that the estimation of optical flow based on the RGB data does not correlate well with the actual movement of objects in the image. Therefore, we sometimes get invalid motion vectors which deteriorate the estimated history of depth values. In particular, we observe that fast movements still cause slight ghosting artifacts, especially for bigger parameter values of $n$.

Our current implementation runs on the GPU and allows to achieve 10 frames per second. At least in case of average temporal filtering, this speed is basically independent of the choices for the parameters $m$ and $n$, which only affect memory consumption and latency. For other filter kernels, which do not allow for an incremental update, the performance will also depend on $n$.

## 5 CONCLUSION

In this work, we have introduced a new strategy to enhance the quality of depth images using optical flow estimated by the corresponding color images. We have tested our approach with different datasets and presented visually appealing results. It remains future work to fine-tune the method for its full potential by evaluating different parameters $m$ and $n$ and higher order temporal filters. It would also be nice to consider longer histories and even preview to some extend. However, in this case the small errors which build up over time would also have an increased impact. Therefore, to address this problem, we will consider a Gaussian filter which levels the impact of history and preview depending on the temporal distance to the current frame. Aside from that, we plan to combine our new method with other refinement strategies. For instance, we consider to include a spatial filtering into our temporal approach for a more robust enhancement.

## 6 REFERENCES

[Brox et al., 2004] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV*, chapter High Accuracy Optical Flow Estimation Based on a Theory for Warping, pages 25–36. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Camplani and Salgado, 2012a] Camplani, M. and Salgado, L. (2012a). Adaptive spatio-temporal filter for low-cost camera depth maps. In *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*.

[Camplani and Salgado, 2012b] Camplani, M. and Salgado, L. (2012b). Efficient spatio-temporal hole filling strategy for kinect depth maps. volume 8290, pages 82900E–82900E–10.

[Camplani and Salgado, 2014] Camplani, M. and Salgado, L. (2014). Background foreground segmentation with rgb-d kinect data: An efficient combination of classifiers. *Journal of Visual Communication and Image Representation*, 25(1):122 – 136. Visual Understanding and Applications with RGB-D Cameras.

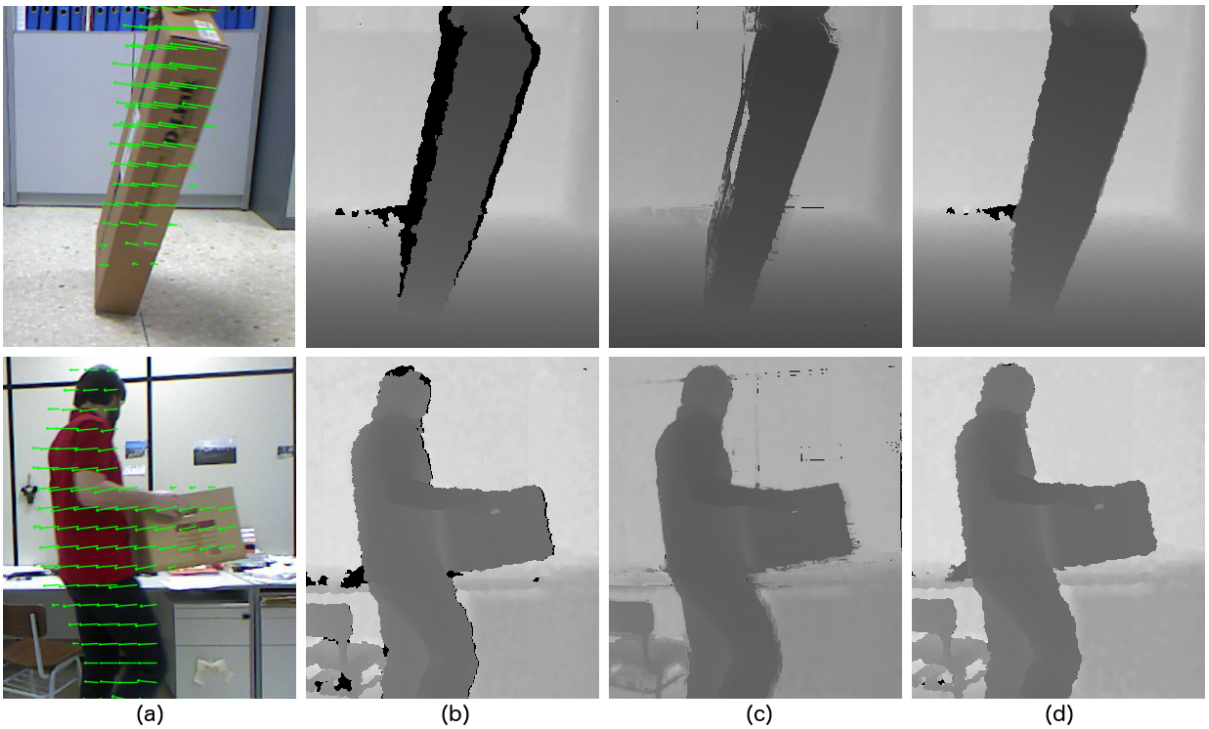


Figure 4: Results for test datasets from [Camplani and Salgado, 2014] - (a) optical flow obtained from the color images, (b) raw depth images (c) output by method from [Garcia et al., 2013] (d) our result. Notably, our results are better for the dynamic parts of the scene.

[Chen et al., 2012] Chen, L., Lin, H., and Li, S. (2012). Depth image enhancement for kinect using region growing and bilateral filter. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3070–3073.

[Garcia et al., 2013] Garcia, F., Aouada, D., Solignac, T., Mirbach, B., and Ottersten, B. (2013). Real-time depth enhancement by fusion for rgb-d cameras. *Computer Vision, IET*, 7(5):1–11.

[Hui and Ngan, 2014] Hui, T.-W. and Ngan, K. N. (2014). Motion-depth: Rgb-d depth map enhancement with motion and depth in complement. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3962–3969.

[Islam et al., 2015] Islam, A. T., Scheel, C., Pajarola, R., and Staadt, O. (2015). Robust enhancement of depth images from kinect sensor. In *Virtual Reality (VR), 2015 IEEE*, pages 197–198.

[Kim et al., 2010] Kim, S.-Y., Cho, J.-H., Koschan, A., and Abidi, M. (2010). Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2358–2361.

[Matyunin et al., 2011] Matyunin, S., Vatolin, D., Berdnikov, Y., and Smirnov, M. (2011). Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4.

[Radford and Burton, 1978] Radford, J. and Burton, A. (1978). *Thinking in perspective : critical essays in the study of thought processes / edited by Andrew Burton and John Radford.* Methuen London.

[Yang et al., 2013] Yang, Q., Ahuja, N., Yang, R., Tan, K.-H., Davis, J., Culbertson, B., Apostolopoulos, J., and Wang, G. (2013). Fusion of median and bilateral filtering for range image upsampling. *Image Processing, IEEE Transactions on*, 22(12):4841–4852.