High-Quality Kinect Depth Filtering for Real-Time 3D Telepresence

Mengyao Zhao[1,2], Fuwen Tan[2], Chi-Wing Fu[1], Chi-Keung Tang[3], Jianfei Cai[1], Tat Jen Cham[1]

[1]School of Computer Engineering & [2]BeingThere Centre, Institute of Media Innovation
Nanyang Technological University, Singapore
& [3]Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

Introduction

3D telepresence is a next-generation multimedia application offering remote users an immersive video-conferencing environment. Kinect sensor, a consumer-grade range camera, facilitates the implementation of some recent 3D telepresence systems. However, when employing Kinect in 3D telepresence applications, we have to consider the following vital issues:

- Raw Kinect depth data is noise-prone and unstable. We need proper filtering to improve the visual quality of 3D data in the 3D telepresence applications;
- Existing data filtering methods on Kinect depth data mainly address uniformly-distributed random noise, but ignore depth quantization problem in the data;
- In 3D telepresence applications, e.g., tele-conferencing, the hardware depth sensors have to be mounted on a fixed location in the physical space, rather than being freely movable as in KinectFusion;
- The filtering process should run in real-time, so we cannot afford tedious computation with global data optimization, e.g., solving with Poisson equations.

Considering above requirements, we propose a multi-scale direction-aware filtering method, aiming at making use of Kinect in 3D telepresence applications. Our method not only can effectively filter Kinect depth data, in particular to resolve the depth quantization issue, but also can run in real-time by using GPUs.

Approach

Kinect Raw Depth Data

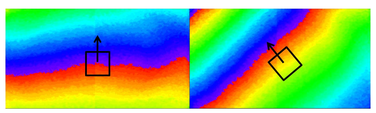
Kinect raw depth data is heavily quantized because of Kinect sensor's low resolution in depth range. One could observe large irregularly-shaped patches of pixels that receive the same depth values from Kinect (Fig.1).

Multi-Scale Filtering

Fig.2 depicts the problem of quantization with a 2D example. Since image features usually occur at multiple scales, the optimal scale to analyze different pixels can be different subject to the local image context.

Direction-Aware Filtering

As in Fig.1, depth quantization orientation is usually not aligned with screen axes because it depends on the angle between the object surface and Kinect viewing direction. Therefore, we attempt to improve the filtering quality by adjusting the pixel weight contributions based on the orientation of local depth gradient.



(a) Kinect's view ⊥ wall (b) Kinect's view slanted

Fig. 1 Color-coded Kinect depth data on a flat wall.

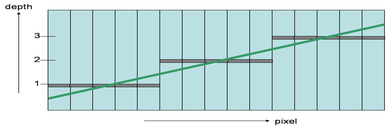


Fig. 2 1D quantized depth with ground truth (green).

Processing Pipeline

1. **Multi-Scale Analysis**: We compute the optimal filter window size (scale) for each pixel in depth map based on a statistical analysis around the pixel neighborhood, and output an optimal scale map.
2. **Direction-Aware Analysis**: We estimate the local predominant orientation by using a structure tensor at each pixel based on an optimal scale (from the previous step) and output an eigenvector map.
3. **Data Filtering**: The two maps are then employed to reconstruct and hole-fill the raw depth map.

Fig. 3 Our Data Processing Pipeline.

Illustration

Our method determines appropriate filtering window sizes for different regions in the same depth map adaptively (see Fig.4 3rd row), and thus can produce higher-quality depth filtering results.

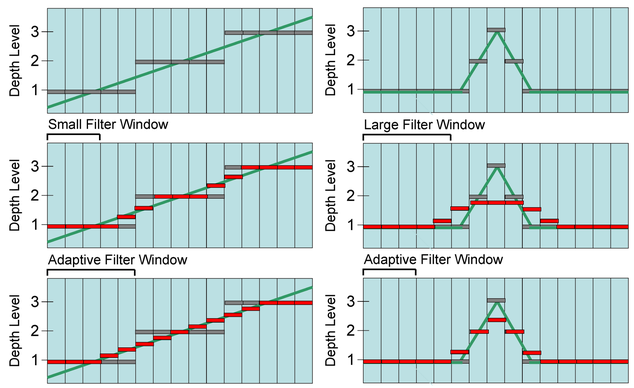


Fig.4 1st row: two different regions in the same raw depth data: large patches (left) and a small feature (right). 2nd Row: applying small (left) and large (right) bilateral filtering windows are insufficient. 3rd row: our multi-scale filtering method avoids producing bumpy patches and can better preserve small surface details. Green lines show the ground truth while red pixels show the filtered result.

Implementation

Algorithm

1. For each pixel, say p, we compute its depth similarity, say f_d, against its neighbor pixel q by

$$f_d(p,q) = e^{-\frac{1}{2}\left(\frac{|D(q)-D(p)|}{\sigma_d}\right)^2}, \quad (1)$$

2. We define function f_e, to measure the directional closeness between pixel p and its neighbor pixel q by

$$f_e(p,q) = e^{-\frac{1}{2}\left(\frac{min\,(v\cdot e_1, v\cdot e_2)}{\sigma_e}\right)^2}, \quad (2)$$

3. Lastly, we filter depth map D by

$$D'(p) = \frac{1}{W(p)} \sum_{q\in\Omega(p)} D(q)\, f_d(p,q)\, f_e(p,q), \quad (3)$$

CUDA Implementation on GPU

We develop a GPU-based method with CUDA to enable real-time depth filtering.

Evaluation & Results

Data Set	Performance on GPU (milliseconds)	
	Bilateral Filter	Our Method
sofa	2.9961	9.0480
ball and box	3.0492	9.7417
human	3.0338	10.171
cloth	3.0634	8.6158

Fig. 5 Performance Evaluation Results.

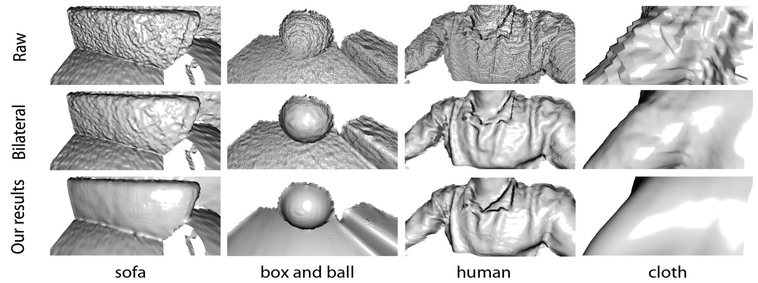


Fig. 6 Our Results.

Conclusion

This paper presents a novel filtering method tailored for filtering raw 3D depth data acquired from Kinect, effectively addressing the depth quantization problem. Experimental results show that surfaces reconstructed from our method are of higher visual quality as compared to the bilateral filtering (Fig.6). CUDA implementation of our method can efficiently run in real-time to support our targeted 3D telepresence application (Fig. 5).

Acknowledgment

This research, which is carried out at BeingThere Centre, is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.