Real-time Preprocessing for Dense 3-D Range Imaging on the GPU: Defect Interpolation, Bilateral Temporal Averaging and Guided Filtering

J. Wasza [1], S. Bauer [1], J. Hornegger [1,2]

[1] Pattern Recognition Lab, Dept. of Computer Science
[2] Erlangen Graduate School in Advanced Optical Technologies (SAOT)
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Range Imaging (RI) Technology

Dense metric 3-D surface acquisition
- 300k points (Kinect, Xtion)
- 40k points (ToF)

Additional Photometric Information
- RGB (Kinect)
- Amplitudes (ToF)

Fig.1: Modern RI devices. PMD CamCube, Asus Xtion and Microsoft Kinect.

Real-time preprocessing is a prerequisite for practical applications due to systematic errors, sensor noise and unreliable measurements!

Real-time Streaming Capability
- 30 fps (Kinect, Xtion)
- 40 fps (ToF)

Broad Range of Applications
- Consumer electronics
- Augmented reality
- Medical engineering

Real-time RI Preprocessing on the GPU

Observation: erroneous range measurements can be grouped into categories: (i) missing or invalid information, (ii) temporal noise and (iii) sensor noise or quantization issues.

Contribution: GPU implementation of a modality-independent and generic pre-processing pipeline (see Fig.2) using the CUDA architecture.

Motivation: compared to a CPU, a much larger portion of GPU resources is devoted to data processing than to caching or flow control (massively parallel architecture).

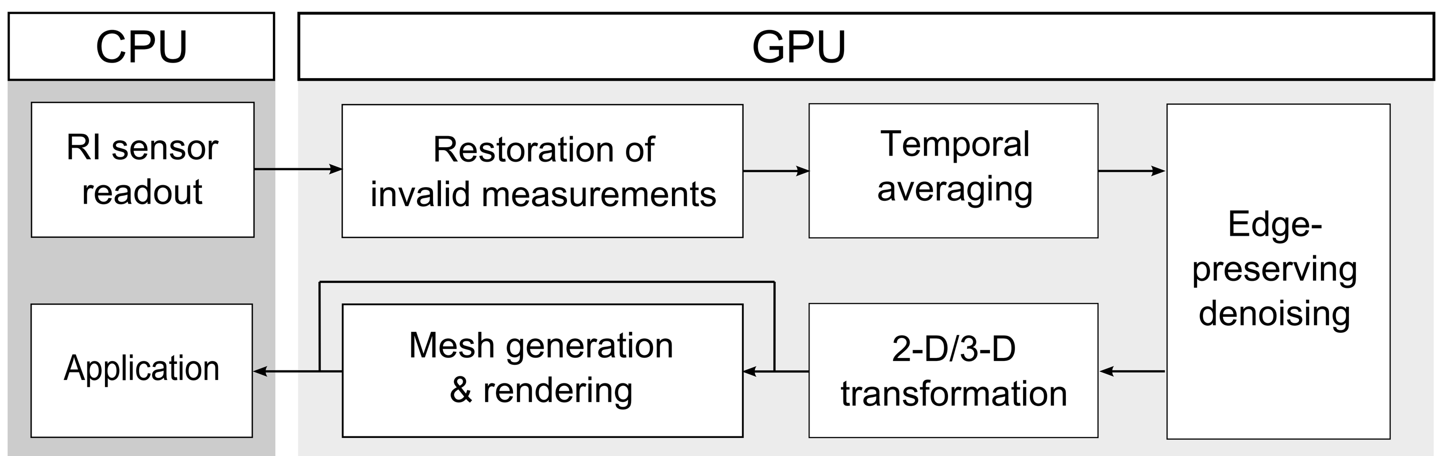

Fig.2: The proposed modality-independent RI preprocessing pipeline.

Methods

Normalized Convolution [1]: restoration of invalid depth measurements

$$f_{\mathrm{NC}}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} g(\boldsymbol{x}')\ a(\boldsymbol{x},\boldsymbol{x}')\ m(\boldsymbol{x}')}{\sum_{\boldsymbol{x}' \in \omega_{\boldsymbol{x}}} a(\boldsymbol{x},\boldsymbol{x}')\ m(\boldsymbol{x}')},$$

$a(\boldsymbol{x},\boldsymbol{x}')$ Applicability function
$m(\boldsymbol{x})$ Validity mask

Bilateral Temporal Averaging [2] : temporal noise suppression in dynamic scenes

$$f_{\mathrm{TA}}(\boldsymbol{x}) = \frac{\sum_{i=0}^{T-1} g_i(\boldsymbol{x}) c(i) s(g_0(\boldsymbol{x}) - g_i(\boldsymbol{x}))}{\sum_{i=0}^{T-1} c(i) s(g_0(\boldsymbol{x}) - g_i(\boldsymbol{x}))},$$

$s(g_0(\boldsymbol{x}) - g_i(\boldsymbol{x}))$ Range similarity
$c(i)$ Temporal closeness

Guided Filtering [3]: O(1) edge-preserving denoising

$$f_{\mathrm{GF}}(\boldsymbol{x}) = \mathrm{E}_{\omega_{\boldsymbol{x}}}(A(\boldsymbol{x})) \cdot i(\boldsymbol{x}) + \mathrm{E}_{\omega_{\boldsymbol{x}}}(B(\boldsymbol{x})),$$

$i(\boldsymbol{x})$ Guidance image
$\mathrm{E}_{\omega_{\boldsymbol{x}}}$ Expectation value

$$A(\boldsymbol{x}) = \frac{\mathrm{Cov}_{\omega_{\boldsymbol{x}}}(i(\boldsymbol{x}), g(\boldsymbol{x}))}{\mathrm{Var}_{\omega_{\boldsymbol{x}}}(i(\boldsymbol{x})) + \epsilon} \qquad B(\boldsymbol{x}) = \mathrm{E}_{\omega_{\boldsymbol{x}}}(g(\boldsymbol{x})) - A(\boldsymbol{x}) \cdot \mathrm{E}_{\omega_{\boldsymbol{x}}}(i(\boldsymbol{x}))$$

$g(\boldsymbol{x})$: corrupted input data
$f(\boldsymbol{x})$: restored and denoised output
$\boldsymbol{x}$: pixel in the 2-D sensor domain
$\omega_{\boldsymbol{x}}$: local quadratic neighborhood of $\boldsymbol{x}$

Experiments and Results

Fig.3: Raw (left) and preprocessed (right) triangulated Kinect data with RGB overlay.

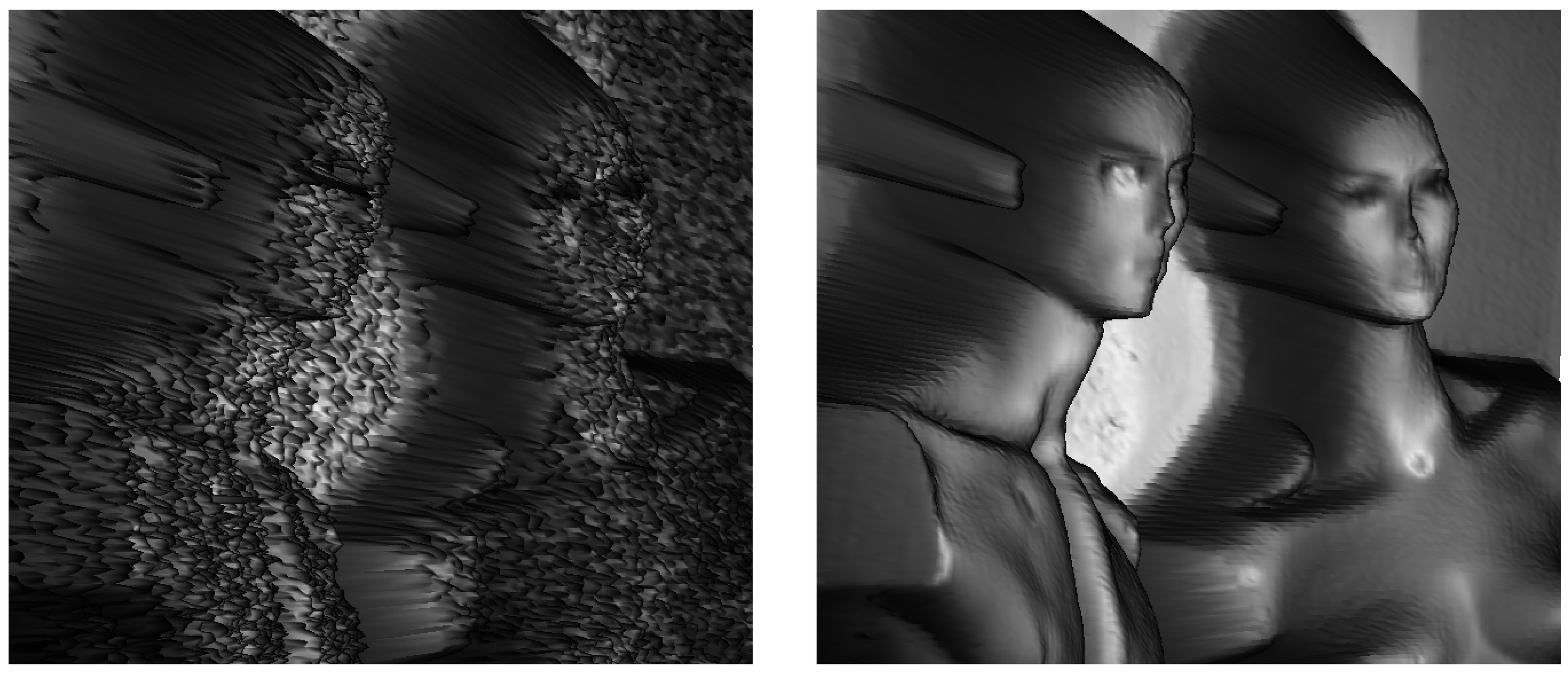

Fig.4: Raw (left) and preprocessed (right) triangulated ToF data with amplitude overlay.

Step	Kinect [ms]	Kinect [percentage]
Host/Device Transfer	1.09 ± 0.07	10.7%
Normalized Convolution	4.00 ± 0.06	39.2%
Bilateral Temporal Averaging	1.49 ± 0.04	14.6%
Guided Filtering	1.93 ± 0.21	18.9%
2-D/3-D Transformation	0.34 ± 0.01	03.3%
Device/Host Transfer	1.39 ± 0.02	13.6%
Total	**10.2**	**100%**

Tab.1: Run-time performance evaluation for Kinect data (640x480 px).

Fig.5: GeForce GTX 285 GPU used in this work.

Step	CamCube [ms]	CamCube [percentage]
Host/Device Transfer	0.23 ± 0.01	05.4%
Normalized Convolution	1.88 ± 0.05	44.2%
Bilateral Temporal Averaging	0.48 ± 0.04	11.3%
Guided Filtering	0.97 ± 0.01	22.8%
2-D/3-D Transformation	0.20 ± 0.03	04.7%
Device/Host Transfer	0.49 ± 0.07	11.5%
Total	**4.25**	**100%**

Tab.2: Run-time performance evaluation for ToF data (200x200 px).

Conclusions

- RI preprocessing can be performed at ~100 fps (Kinect) and >200 fps (ToF) on a consumer GPU
- Low run-times allow to devote GPU resources to postprocessing and are promising w.r.t future RI sensors
- Alternative real-time capable preprocessing modules will be subject of our upcoming research
- Preprocessing plugins available for the *Range Imaging Toolkit* [4] at **http://www5.cs.fau.de/ritk**

RITK THE RANGE IMAGING TOOLKIT

Acknowledgments: We gratefully acknowledge the support by the European Regional Development Fund (ERDF) and the Bayerisches Staatsministerium für Wirtschaft, Infrastruktur, Verkehr und Technologie (StMWIVT) under Grant No. IUK338/001.

References

[1] H. Knutsson and C.-F. Westin. Normalized and Differential Convolution: Methods for Interpolation and Filtering of Incomplete Data. In *Proc. IEEE CVPR 1993, 512-523*

[2] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In Proc. *IEEE ICCV 1998*, 839-846

[3] K. He, J. Sun and X. Tang. Guided Image Filtering. In *Proc ECCV 2010: Part I, vol. 6311*, 1-14

[4] J. Wasza et al. RITK: The Range Image Toolkit - A Framework for 3-D Range Image Stream Processing. In *Proc. VMV 2011*, 57-64