High Performance GPU-based Preprocessing for Time-of-Flight Imaging in Medical Applications

Jakob Wasza[1], Sebastian Bauer[1], Joachim Hornegger[1,2]

[1]Pattern Recognition Lab, Friedrich-Alexander University Erlangen-Nuremberg
[2]Erlangen Graduate School in Advanced Optical Technologies (SAOT)
jakob.wasza@informatik.uni-erlangen.de

Abstract. Time-of-Flight (ToF) imaging is a promising technology for real-time metric surface acquisition and has recently been proposed for a variety of medical applications. However, due to limitations of the sensor, range data from ToF cameras are subject to noise and contain invalid outliers. In this paper, we discuss a real-time capable framework for ToF preprocessing in a medical environment. The contribution of this work is threefold. First, we address the restoration of invalid measurements that typically occur with specular reflections on wet organ surfaces. Second, we compare the conventional bilateral filter with the recently introduced concept of guided image filtering for edge preserving denoising. Third, we have implemented the pipeline on the graphics processing unit (GPU), enabling high-quality preprocessing in real-time. In experiments, the framework achieved a depth accuracy of 0.8 mm (1.4 mm) on synthetic (real) data, at a total runtime of 40 ms.

1 Introduction

Recent advances in Time-of-Flight (ToF) imaging have opened new perspectives for its use in medical engineering. In particular, the resolution (40k points), frame rate (40 Hz) and validity information provided by the camera hold potential for medical applications. ToF imaging has, among others, been proposed for 3D endoscopy [1] and intra-operative organ surface registration [2]. The inherent high degree of accuracy for these tasks requires a preprocessing pipeline to cope with the noisy and corrupted data obtained from the ToF sensor. Even though a proper sensor calibration [3] can be used to eliminate systematic errors, denoising provides the fundamental basis to produce steady and reliable surface data. At this, temporal averaging and edge preserving filters are commonly used [2,3]. One issue that cannot be addressed by conventional filters is the elimination of invalid depth values caused by specular reflections that lead to saturated sensor elements. These effects often occur with ToF surface acquisition of organs due to their shiny and wet surface. In this paper, we propose to use a spectral domain method known from digital radiography to estimate the depth information at invalid pixels. As low filter runtimes are a crucial factor, we investigate the performance and robustness of the bilateral filter and the recently introduced guided image filter for edge preserving denoising. To ultimately achieve real-time capability, we implemented all filters on the graphics processing unit (GPU).

2 Materials and Methods

2.1 Preprocessing pipeline

The preprocessing pipeline in this work consists of three modules that operate on the distance information: (i) defect pixel interpolation, (ii) temporal averaging, (iii) edge preserving denoising.

Defect pixel interpolation In order to correct invalid depth measurements, we adopt a spectral domain method that was proposed by Aach and Metzler for defect pixel interpolation in digital radiography [4]. The basic assumption is that the observed corrupted signal g can be expressed as a multiplication of the ideal signal f with a binary defect mask w given by the validity information provided by the camera. This corresponds to a convolution $(*)$ in the frequency domain:

$$g = f \cdot w \;\equiv\; G = F * W \tag{1}$$

where F,G and W denote the spectra of f, g and w, respectively. The unknown complex coefficients of F are then estimated by an iterative spectral deconvolution scheme and the restored ideal signal is obtained as:

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + (1 - w(\boldsymbol{x})) \cdot \widehat{f}(\boldsymbol{x}) \tag{2}$$

where $\widehat{f}$ denotes the inverse Fourier transform of the estimated spectrum F.

Temporal averaging For a frame at time t we perform temporal denoising by computing the arithmetic mean of N successive frames g_i:

$$f_t(\boldsymbol{x}) = \frac{1}{N} \sum_{i=t-N+1}^{t} g_i(\boldsymbol{x}) \tag{3}$$

We note that this filter can be implemented in a recursive manner, i.e.:

$$f_t(\boldsymbol{x}) = \frac{1}{N} \left(N \cdot f_{t-1}(\boldsymbol{x}) - g_{t-N}(\boldsymbol{x}) + g_t(\boldsymbol{x}) \right) \tag{4}$$

This formulation provides an effective way to reduce GPU memory usage as not all N frames have to be stored and accessed. As the evaluation of equation (4) per pixel $\boldsymbol{x}$ can be executed in parallel, an implementation on the GPU is straightforward.

Edge preserving denoising The bilateral filter proposed by Tomasi and Manduchi [5] is a very popular spatial denoising filter in ToF imaging. The discrete version for the Gaussian case can be expressed as:

$$f(\boldsymbol{x}) = \frac{1}{K_{\boldsymbol{x}}} \sum_{\boldsymbol{y} \in \omega_{\boldsymbol{x}}} g(\boldsymbol{y}) \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|_2}{\sigma_s^2} \right) \exp\left(-\frac{|g(\boldsymbol{x}) - g(\boldsymbol{y})|}{\sigma_r^2} \right) \tag{5}$$

where g denotes the input image and $\omega_{\boldsymbol{x}}$ denotes a local window centered at coordinate $\boldsymbol{x}$. $K_{\boldsymbol{x}}$ is a normalization factor, σ_s and σ_r control the spatial and range similarity, respectively. Due to its translational-variant kernel this filter is computationally expensive. Nevertheless, it can be implemented efficiently on the GPU as the evaluation of equation (5) for all pixels $\boldsymbol{x}$ can be performed concurrently. In order to cope with boundary conditions and potentially uncoalesced GPU memory access patterns the input image g is bound as a texture.
Recently, the guided image filter was proposed by He et al. [6]. This filter has a non-approximative linear-time algorithm for edge preserving filtering, thus, being very promising for real-time ToF preprocessing. The filter output for a pixel $\boldsymbol{x}$ can eventually be deduced as:

$$f(\boldsymbol{x}) = \left(\frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{y} \in \omega_{\boldsymbol{x}}} a_{\boldsymbol{y}} \right) I(\boldsymbol{x}) + \frac{1}{|\omega_{\boldsymbol{x}}|} \sum_{\boldsymbol{y} \in \omega_{\boldsymbol{x}}} b_{\boldsymbol{y}} \tag{6}$$

where $\omega_{\boldsymbol{x}}$ denotes a local window centered at $\boldsymbol{x}$ and I denotes the guidance image. The evaluation of the summations in equation (6) and the estimation of the coefficients $a_{\boldsymbol{y}}$ and $b_{\boldsymbol{y}}$ (see [6]) can be done by using box filters. In turn, box filtering can be performed efficiently by using integral images which provides the basis for a linear-time algorithm. We employ the *parallel-prefix-sum* algorithm as described in [7] for the computation of integral images on the GPU.

2.2 Experiments

In order to assess the accuracy of the presented methods we evaluate the absolute distance error between a ground truth and a template object on a per-pixel basis.

Synthetic data Deciding on a ground truth for the evaluation is a non-trivial task as the ideal metric surface of the observed object is in general not known. Therefore, we conduct experiments on simulated distance values reconstructed from the z-buffer representation of a 3D scene. These values constitute the unbiased ground truth in this experiment. We then approximate the temporal noise on a per-pixel basis by adding an individual offset drawn from a normal distribution with $\sigma = 10$ mm and $\mu = 0$ mm. This standard deviation is motivated by observations on real ToF data. In order to simulate the effect of amplitude related noise variance and total reflections we additionally corrupt the distance data by Perlin noise [8]. For this study, we rendered a porcine liver mesh segmented from a CT scan.

Real data The focus of the real data experiments is the comparison of a common preprocessing pipeline (temporal averaging, edge preserving denoising) with a pipeline that additionally performs defect pixel interpolation. We conducted the experiment using a PMD CamCube 3.0 with a resolution of 200×200 pixels. For the ground truth generation, we averaged 1000 frames acquired from an uncorrupted liver phantom and applied a bilateral filter with $\sigma_s = 50$ mm and

Table 1. Error analysis on synthetic data and filter runtimes.

Filter	None	DPI	TA	BF	GF
Mean error [mm]	6.1 ± 8.2	5.5 ± 5.1	1.8 ± 1.5	0.4 ± 0.6	0.8 ± 2.1
Runtime [ms]	n/a	34 ± 6.3	0.7 ± 0.1	2.9 ± 0.1	2.6 ± 0.2

$\sigma_r = 5$ pixels to smooth out systematic artifacts that could not be corrected by sensor calibration. We then placed two pieces of aluminum foil onto the phantom to simulate a reflective surface. Using this corrupted phantom, we assessed the results of the standard preprocessing pipeline as applied for the ground truth and the same pipeline performing a defect pixel interpolation as a first step.

3 Results

We have implemented the defect pixel interpolation (DPI), temporal averaging (TA), bilateral filter (BF) and guided image filter (GF) on a Quadro FX 2800M GPU using NVidia's CUDA technology. Qualitative results for the defect pixel interpolation on real data are depicted in Fig. 1. Without correction the corrupted regions show a mean error of 68.4 ± 40.2 mm. Using defect pixel interpolation we were able to reduce this error to 1.4 ± 1.1 mm. Qualitative and quantitative results for the filter evaluation on synthetic data are shown in Fig. 2 and Table 1 whereby the error reduction is cumulative across filters while runtimes are not. Using the presented preprocessing pipeline we were able to reduce the mean error across the surface from 6.1 ± 8.2 mm to 0.4 ± 0.6 mm and 0.8 ± 2.1 mm for the bilateral and the guided image filter, respectively. The bilateral filter shows a better accuracy at very strong edges, see Fig. 2. Given a total pipeline runtime of approximately 40 ms, real-time constraints are satisfied.

4 Discussion

We have presented a real-time capable preprocessing pipeline for ToF imaging in medical applications. The defect pixel interpolation yielded promising

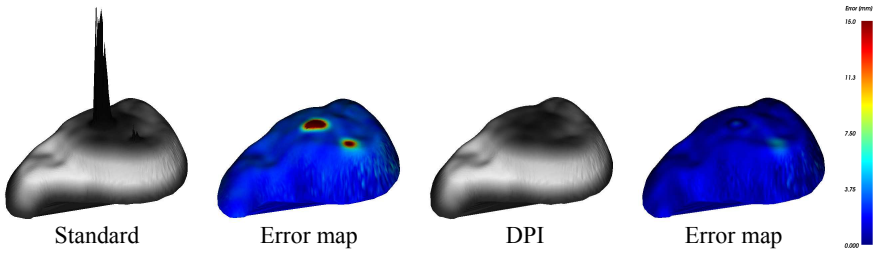
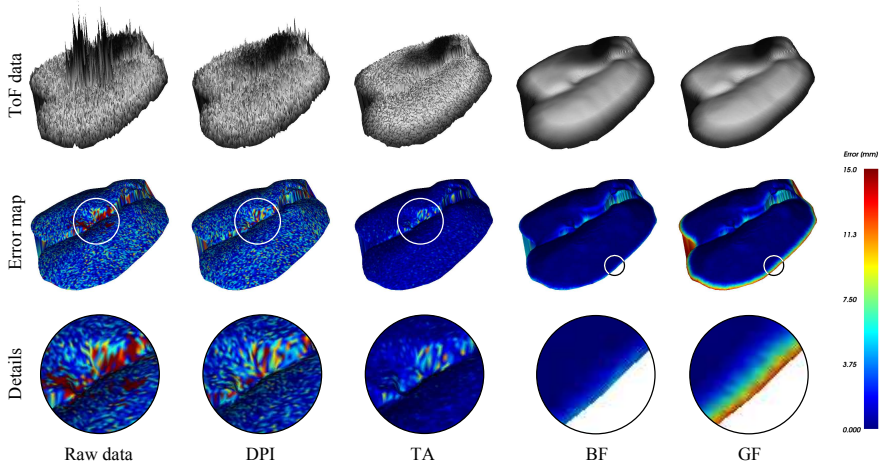


Fig. 1. Defect pixel interpolation on real data.

Fig. 2. Filter operations on synthetic data.



results with regard to accuracy. Nonetheless, future research has to investigate alternative spectral domain methods as well as spatial domain methods for the restoration of invalid depth values. For edge preserving denoising, guided image filtering turned out to be an alternative to the bilater filter. However, the latter shows a better behavior at sharp edges. Future work has to investigate adaptive variants of edge preserving filters that additionally account for the amplitude related noise variance. Concerning runtime issues, we note that even on a current mid-range GPU real-time constraints can be satisfied. This is a promising result with regard to next-generation and potentially high-resolution ToF sensors.

References

1. Penne J, Höller K, Stürmer M, Schrauder T, Schneider A, Engelbrecht R, et al. Time-of-flight 3-D endoscopy. In: Proc MICCAI. vol. 5761; 2009. p. 467–474.
2. Seitel A, Santos T, Mersmann S, Penne J, Tetzlaff R, Meinzer HP. Time-of-Flight Kameras für die intraoperative Oberflächenerfassung. In: Proc BVM; 2010. p. 11–15.
3. Lindner M, Schiller I, Kolb A, Koch R. Time-of-Flight sensor calibration for accurate range sensing. Comput Vis Image Underst. 2010; p. in press.
4. Aach T, Metzler V. Defect interpolation in digital radiography - how object-oriented transform coding helps. In: Proc SPIE. vol. 4322; 2001. p. 824–835.
5. Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: Proc ICCV; 1998. p. 839–846.
6. He K, Sun J, Tang X. Guided image filtering. In: Proc ECCV. vol. 6311; 2010. p. 1–14.
7. Harris M, Sengupta S, Owens JD. Parallel Prefix Sum (Scan) with CUDA. In: Nguyen H, editor. GPU Gems 3. Addison Wesley; 2007. .
8. Perlin K. An image synthesizer. SIGGRAPH Comput Graph. 1985;19(3):287–296.