GPU-Accelerated Adaptive Image Denoising

Konstantin Sushchyk

Abstract - **In this paper an efficient GPU-based implementation of adaptive bilateral filtration is considered.**

Keywords – **Adaptive noise removal, bilateral filtration, GPGPU, CUDA**

I. Introduction

Despite of a great progress in an image acquision hardware image denoising still remains an important part of image processing. Removing of a noise does not only improves human image perception but is required as a preliminary step by a number of image processing applications i.e. machine vision and pattern recognition. Moreover, a size of images continues to grow as a resolution of digital sensors increase, that lead to need for efficient noise removal algorithms.

II. Bilateral filtration

Conventional linear filter works well in smooth regions, however it substantially blurs the edges of an image. A bilateral filter proposed by C. Tomasi and R. Manduchi [1] is an edge-preserving and noise reducing smoothing filter. As the most of other filters this filter replaces the intensity value at each pixel in an image with a weighted average of intensity values from a pixels in some neighborhood. However, the weight depends not only on Euclidian distance, but also on difference in the intensity range, that allows to preserve sharp edges, while removing a noise on smooth image areas.

A denoised pixel value for the current pixel x is computed as

$$h(x) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)\, c(x,x)\, s(f(x), f(x))\, dx}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x,x)\, s(f(x), f(x))\, dx},$$

where $f(x)$ is a noised value for the current pixel, $f(x)$ is a value from the neighborhood, $c(x,x)$ is a domain filter and $s(f_x, f_x)$ is a range filter.

B. Zhang and J.P. Allebach proposed a generalization of bilateral filter allowing its parameters to be locally adaptive[2]. The parameter optimization is formulated as a Minimum Mean Squared Error (MMSE) estimation problem. This approach outperforms the bilateral filter in noise removal and edge preserving. Contrary to the approach proposed in [2] more general domain filter has been used in this work. Range filter, in turn, has been simplified.

However it is a computationally expensive algorithm that requires a selection of a range filter parameters for each pixel of the image. Computational complexity together with growing image size leads to the fact that CPU is not suitable for this task anymore.

III. GPGPU and Nvidia CUDA technology

Graphics Processing Units (GPUs) are used rendering for and computer graphics acceleration. However, they are essentially a SIMD processing units consisting of parallel computational elements accessible with some effort for arbitrary, not necessary graphical computation. This approach of solving general-purpose problems on GPUs is known as general-purpose computing on graphics processing units (GPGPU). The model for GPU computing is to use a CPU and GPU together in a heterogeneous co-processing computing model where sequential part of the application runs on the CPU and the computationally-intensive part is accelerated by the GPU [3].

CUDA (Compute Unified Device Architecture) is a new massively parallel GPGPU solution from Nvidia, that consists from hardware architecture and parallel programming model. Nvidia provides C++ compiler with syntax extensions for paralellization, high-level libraries for solving linear systems (CUBLAS), data manipulation(Thrust, CUDPP) signal and image processing (cuFFT, NPP). Language bindings to a number of languages including MATLAB, Python, Java and Ruby are provided by Nvidia and third parties as well.

IV. Conclusion

In this work a classical bilateral filter as well as modified adaptive bilateral filtration has been implemented using Nvidia CUDA architecture and NPP library. Adaptive technique provides better visual quality and SNR than classical bilateral filtering proposed in [1]. A use of GPU allows to achieve substantial acceleration. Experiments using Nvidia GTS–450 GPU card confirm that GPU implementation outperforms CPU implementation by up to a factor of twenty.

References

[1] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images", Proceedings of IEEE International Conference on Computer Vision '98, New Delhi, January 1998, pp. 836-846.

[2] B. Zhang, J.P. Allebach, “Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal” IEEE Trans. Image Process., vol. 17, No. 5,2008, pp. 664-678.

[3] W. Ramey. “Languages, APIs and Development Tools for GPU Computing”, GPU Technology Conference, 2010