Image Processing by using Bilateral Filtering with CUDA: A Review

Ashish A. Shastrakar
Student, M.Tech (Electronics Engineering (communication))
Electronics and communication Engineering
Vidharbha Institute Of Technology, Nagpur, India
ashishastrakar@gmail.com

Prof. Nilesh P. Bodne
Research Scholar
Electronics and Communication Engineering
Vidharbha Institute Of Technology, Nagpur, India
nileshbodne@gmail.com

Abstract — **A digital image is a representation of a 2-dimensional image as a finite set of digital values, called picture elements or pixels. Bilateral filtering smooth images while maintaining edges by using of non-similar combination of nearing images value. The method is less time consuming easy and local. High resolution video capture device and increase requirement for accuracy make it even harder to realise real time performance. NVIDIA is a currently design interface providing a CUDA (compute unified device architecture). Production chain futures in industrial vision are becoming more and wide spread. Represent the result on graphical users interference.**

INDEX TERMS—— BILATERAL FILTERING, IMAGE PROCESSING,GRAPHICAL USER INTERFACE, CUDA

I.INTRODUCTION

The objective of this project is to create and implement compute unified device architecture written module in a visual growth environment.

The purpose is to accelerate the already existing image processing module by dispatching a part of the process to the graphical processing unit. The application can be used for the medical, industrial and artistic purpose.

Image blurring is widely used effect in graphics software, typically to reduce image detail and reduce image noise. The visual effect of this blurring technique is not rough blur resembling that viewing the image through a translucent screen, distinctly separate from bokeh effect create by a shadow of an object under the visual illumination or out of focus lenses.

In the section II, bilateral filter are described. In the section III, system experiments with black & white images are described, in section IV experiment with color images can be described and CUDA are described in section V. The result of this preview paper is explained in section VI and future works are summarized in section VII

.

II.BILATERAL FILTER

For color image filtering is most important. 3bands of color images should not be filtered separately from one another, as color get corrupted close to images edges. In fact, different bands have different contrast level and are smoothed differently. Separate smoothing disturbs the balance of color and undesirable color combination is appearing. Bilateral filters has able to operate the 3 band at once and be told explicitly, so to speak which colors are similar or which are not. Only perceptually same colors are then average together and artificial mentions above disappear.

$$I^{\text{filtered}}(x) = \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

*I*filtered is the filtered image.
*I*is original input image to be filtered.
X are coordinates of current pixel to be filtered.
Ω is window centered in X.
fris range kernel for smoothing differences in intensities.
gris spatial kernel for smoothing differences in coordinates.

The idea of implementation bilateral filtering is working in the range of an image what traditional filters do in the range of domain. 2 pixels are close to one each another i.e. has the approximate value possible exact in perceptually meaningful way. In old days, filtering is a domain and enforces the closeness by weighing of the pixel value with coefficients that fall in with distance.

III. EXPERIMENTS WITH BLACK & WHITE IMAGES

In this part we analyze the improvement of bilateral filter on black & white images figure 1& 2 in the color plates observe the potential of bilateral filtering for the withdrawing of the texture. Less amount of grey level quantization can be observing in fig.2, but this is possible due to printing process, not by the filtering process. Columns to different amount of range filtering and rows correspond to different amount domain filtering.

Figure 1

Figure 2

IV. EXPERIMENTS WITH COLOR IMAGES

For black and white images, intensities between any two grey levels are still grey level. As a sequence when smoothing black and white images with a standard LPF, certain intermediate level of grey are created through the edges, leading to production of blurr images. With color image, an extra problem created from the fact that between two colors there are others, again rather different colors. The filter images are not just look a blurred, it also consist of odd looking, colors around object.

While single repetitions produces clear images figure 4 than the original images, and are likely to be sufficient for more images filtering in needs. Multiple iteration have flattening the colors in an image, but without blur image.

The output image has a much reduce a color map, and effect of bilateral filter are simple to observe when the printout is taken. All shadows and edges are preserve but most of the shading is gone and no new colors are introduce by filtering process.

Figure 3

Figure 4

Figure 5

V. CUDA

The most recently, the new programming model and architecture model has expose has more flexibility on graphics processing unit hardware. CUDA (compute unified device architecture) allow the program to write a C program which no

longer requires dependence for knowledge on graphics pipelining. Faster memory share between group of processor can act as a programmable cache memory, accelerating special local operation.

High level program view of CUDA architectures and mapping techniques are discussed. When dealing with camera data in the form of byte elements, graphics processing unit may perform hardware data conversion from 8 or 16 bit integer value to floating point during the fetch. When fetching, the architecture may sample with bilinear interpolation, useful in stereo matching. Convolution operation at image edges can be made simple by clamping and wrap modes of operation at texture borders.

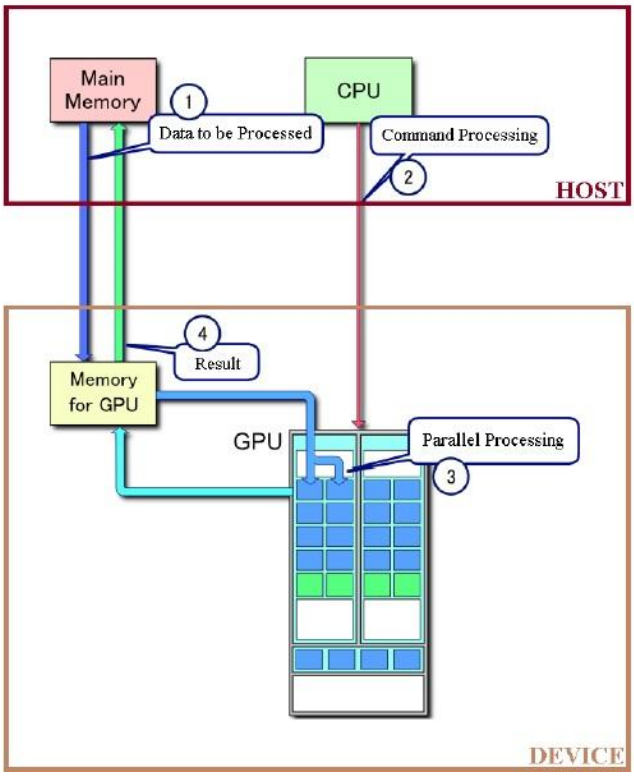

Fig.6 Programming steps in CUDA

In order to obtain good performance parallelizable phases must be executed above the device. In image processing CPU & GPU have important role to play.

- **CPU:** Reading the picture
- **GPU:** Processing
- **CPU:** Action to be done depending on processing to be done
- **CPU:** Memory cleaning

Graphical processing unit has produce from special purpose processor to programmable processor and programmability has been most important feature.

- General programming environment
- More powerful parallel computing capability
- Higher bandwidth
- Instruction operation

VI. RESULT

In this paper we have introduced the concept of bilateral filtering for edge preserving smoothing. The parameter used for bilateral filtering in our taking example were some extending arbitrary. Give specific application, technique for automatic design for filter profile and parameter value may possible.

It was noted that in order to get advantages of GPU, several process are to be proceed and calculations are to be complicated enough.

VII. FUTURE SCOPE

Bilateral filtering for image processing, take considerable amount of time processing an image with loss of information, to decrease this competition time, this technique implemented on CUDA. CUDA on GPU lead to huge amount of reduction in processing time.

VIII. REFERENCE

[1] J. Sanders and E. Kandrot: *CUDA by exemple – An Introduction to General-Purpose GPU Programming*, Addison-Wesley, Michigan, October 2010.
[2] D. B. Kirk and W. W. Hwu: *Programming Massively Parallel Processors – A Hands-on Approach*, Morgan Kaufman, 2009.
[3] J. Kreidler: *Jloadbang – Programming Electronic Music in Pure Data*,Wolke Verlag, Hofheim, 2009.
[4] Tom R. Halfhil, "Parallel Processing With CUDA", Microprocessor Report, Scottsdale, Arizona, Jan 28, 2008.
[5] WU En Hua , "State of the Art and Future Challenge on General Purpose Computation by Graphics Processing Unit", Journal of Software, vol. 15, no. 10, 2004,pp.1493~1504.
[6] Michael Boyer, Kevin Skadron, Westley Weimer. "Automated Dynamic Analysis of CUDA Programs", STMCS 2008, Boston, Massachusetts, Apr 06, 2008.
[7] Shubhabrata Sengupta, Mark Harris, Yao Zhang, and John D. Owens, "Scan Primitives for GPU Computing", Graphics Hardware 2007, San Diego, California, August 04 - 05, 2007.
[8] Samer Al-Kiswany, Abdullah Gharaibeh, Elizeu Santos-Neto, et al.. "StoreGPU: Exploiting Graphics Processing Units

to Accelerate Distributed Storage Systems", HPDC'08, June 23–27, 2008, Boston, Massachusetts, USA.
[9] John D. Owens, Mike Houston, David Luebke, et al., "GPU Computing", Proceedings of the IEEE, vol. 96, no. 5, May 2008 ,pp.879-897.