International Journal of Scientific Research and Management (IJSRM)
||Volume||5||Issue||05||Pages||5336-5340||2017||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47 DOI: 10.18535/ijsrm/v5i5.09

Reducing the Computation Complexity of 2-D Gaussian Filter

Deepak raj[1], Poonam singal[2], and Namika kumari[3]
Deenbandhuchhotu Ram University of Science and Technology, Murthal (Sonipat)

Abstract—Now a day's we are facing a problem regarding smoothing of the image.One of the very useful method to smoothing the image is 2-D Gaussian Filter, which is used in Image Processing. However, the heavy computational resources are required by 2-D Gaussian Filter, and it comes down to real-time applications. The vital efficiency is achieved in this implementation. We use floating-point representation, but there are certain obstacle for this implementation because it requires large computational power in order to achieve real-time image processing. On the other hand a fixed-point approach is much more suitable for the implementation of a 2-D Gaussian filter in FPGA.By using fixed-point arithmetic for the implementation, the efficiency is increases, size of area decreases, complexity decreases and the computational cost is also reduced.In this paper we reduced the LUT by 11.34 % by using Gaussian filter.

Keywords—LUT-Look up table, real-time Gaussian filter, fixed-point, computational cost

1.INTRODUCTION

We are living in digital words. Digital world means technical and modern world. The idea of digital image processing arebeing applied in different fields such as medicine, astronomy, geography, industry, etc. [1]. These fields often require results in real-time, and efficiency in the implementation of digital image processing is imperative. In this paper, wepresent how the LUT is reduced by the implementation of a 2D Gaussian Filter on an FPGA [2].

Image Processing is processing of images using mathematical operations by using any form of processing for which the input is an image,a series of images, or a video. Image processing are basically a digital processing but optical and analog image are also possible [3]. Computer graphics and computer vision are also related to image processing. Images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies [4]. On the other hand computer vision is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans) [5].In this paper we take an example of convolution using FPGA was developed using [3*3] sobel filter using a [15*15] image. [6].

The remaining paper is divided into 4 sections. Section 2 covers the theory, Section3 explains the methodology of the Gaussian filter; shows a performance of the Gaussian filter and Kernel Quantization; while Section 4 presents the hardware implementation for the Gaussian filter for fixed-point. In Section 5, the results of the filter implementations are discussed and give the conclusion.

2. THEORY

I .Role of Filters

DOI: 10.18535/ijsrm/v5i5.09

For increasing the sharpness and brightness of the image we use a filter circuit. To emphasize a desire features or to remove undesired feature we use filter. Any image can be represented by a matrix of pixels withvalues ranging from 0 to 255 [7]. For an example we send a [512 x 512] image to the FPGA requires converting that image into a vector of 2 6 12 444 elements, where DATA is the pixel value and ADDR is the memory address of each pixel [8].

II. Image Smoothing

Smoothing is the process which is used to control the extent of smoothing because it is a tuning parameter. The main purpose of smoothing is to remove the noise or other unwanted phenomena. In smoothing the images are modifying so that the individual points are noise free. There are two important cause in which smoothing are used that can aid in data analysis [9].

III. Sharpening Filters

Image sharpening means to emphasis the shape and drawing the viewer focus. If the high frequencies are attenuated or completely removed, then the visual quality of an image will be poor. In other way we can say that if the high frequency components of an image are used then the quality of the image is improved. Image sharpening means to enhance the highlighted edges and fine details in an image. For effective sharpening process we generally use the high-pass filter operation. Basically linear filter has been used in the implementation of high pass filter [10].

3. METHODOLOGY

I.Gaussian filter

For smooth and noise free image, the Gaussian filter is used which is a 2-D convolution operator. The convolution of an image w(x,y) of size m*n with a function of f(x,y) [10], denoted by w(x,y).f(x,y), is

$$\omega(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x-s, y-t) \quad (1)$$

given by equation (1),

In this equation a and b are given by a = (m-l)/2 and b = (n-1)/2. Figure 1 illustrated the graphical representation of equation (1), Function f (x, y) defines the Gaussian filter, established bythe

$$f(x,y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Gaussian distribution in equation (2):

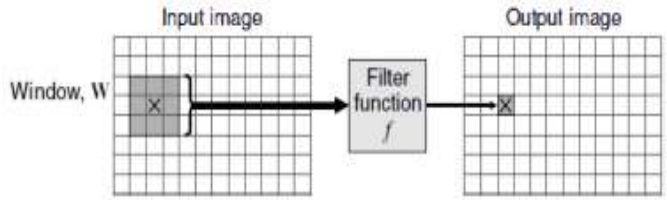


Figure 1. A window filter. The shaded pixels represent the input window that produced the filtered value for the corresponding location in the output image. Each possible window position generates a corresponding location in the output image [8].

The dimensions of the windows and the standard deviation are the two parameters of the Gaussian filter. We required a large standard deviation for the smoothing image.

This filter also reduced the some noise like “salt andPepper” and extracting features on the SALT algorithm [11], salt and pepper noise are shown in figure (2).

Figure2. Image with 30% salt & pepper noise

For the perspective of spatial and frequency domains, Gaussian smoothing filters are low pass filters. These filters are efficient to implement and can be generally used by engineers in practical vision applications [6].

DOI: 10.18535/ijsrm/v5i5.09

II. 2-D Gaussian Filter

There are various Gaussian function like 1D, 2D and ND, whose expression are as follows:

$$G_{1D}(x) = \frac{1}{\sqrt{2\Pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

$$G_{2D}(x,y) = \frac{1}{\sqrt{2\Pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

$$G_{ND}(\vec{x}) = \frac{1}{\sqrt{2\Pi}\sigma} e^{-\frac{|x|^2}{2\sigma^2}} \quad (5)$$

Where width of the Gaussian functions is indicated by σ. If we considered the Gaussian probability density function, then σ is considered as a standard deviation and variance (σ^2) is the square of standard deviation [12].

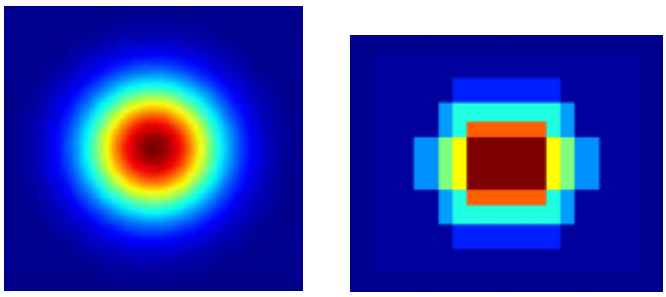
The 2-D Gaussian function in discrete version is shown in equation (6)

$$G_n(x,y) = K_n \times e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (6)$$

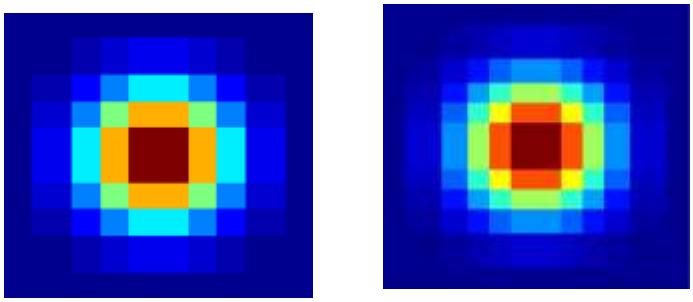
where spatial coordinates are indicated by x and y, where M * N represents the kernel size, n is preferred as 1,2 and 3, if the Gaussian function is designed with three scales, namely, 16, 64, and 128.

III. Kernel Quantization

Gaussian filter is implemented in FPGA by using VHDL description language. In this filter the addition and multiplication are process in between the image and the kernel, where image is represented by a matrix with value 0 to 255 98 bits). The normalized square matrix is kernel, which values are between 0 and 1.



(a)Gaussian kernel (b) SII [24] approximation



(c) Proposedapproximation (4 constant) (d) Proposedapproximation (5constant)

Figure 3. Comparison of (a) exact Gaussian kernel, (b) stacked integral images [13] with 5 2-D boxes, and the proposed method with 4 constants (c) and 5 constants (d). [14]

4. HARDWARE IMPLEMENTATION

Any image can be represented by a matrix of pixels with values ranging from 0 to 255. Sending a [512 x 512] image to the FPGA requires converting that image into a vector of 2 6 12 444 elements as shown in Figure 4, where DATA is the pixel valueand ADDR is the memory address of each pixel, respectively.

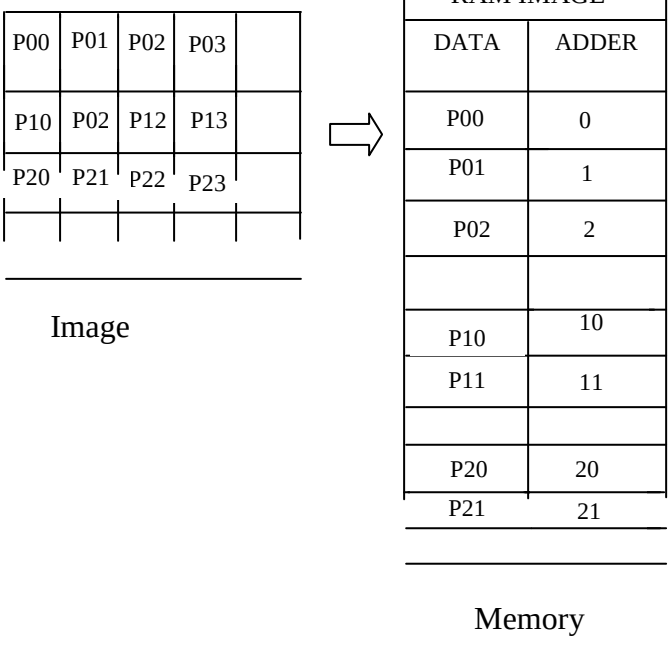


Figure 4. Conversion of an image Matrix to an image vector

Each [3*3] sun image is extracting from the original image, from top to bottom, and from left to right, occurs in the convolution of an image with a [3*3] kernel. After that kernel image are multiplied with

each element, and the resulting products are finally added together (by using multiply-accumulate – MAC- operation), and finally returned to an output pixel. MAC process is explained in figure (6).

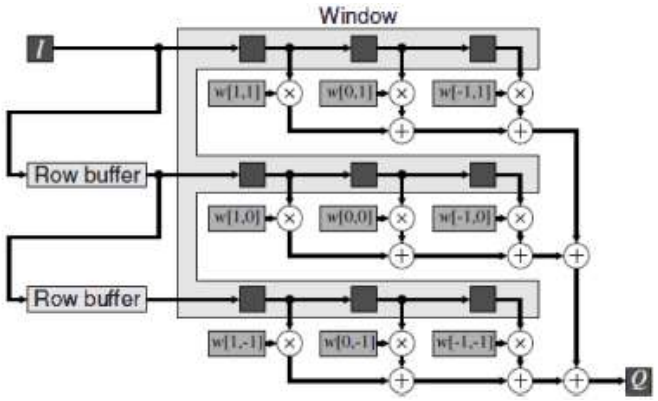


Figure 5. Extraction of a [3*3] sub image and MAC operation [8]

In Figure 5, the row buffer is used to store the rows of pixels belonging to the [3*3] window neighborhood. Shift registers are used to store the pixels from these rows. We required N-1 rows buffers to solve the convolution of N*N. FIR Filtering is same as this process.

Figure 6. Fixed point representation

5. RESULT

The usage of FPGA resources using fixed point method for a [3*3] kernel and a standard deviation 0.5 is taken. As the kernel size increases, the use of LUTs and slice registers will be increases by internal implementations of FIFO memories. In our design we see that thethe implementation of a 2-D Gaussian filter in FPGA.

Table 1: Comparison between with and without Gaussian filter

Resource	Estimation without Gaussian filter	Estimation with Gaussian filter
FF	76	106400
LUT	86	53200
I/O	1	200

By using fixed-point arithmetic for this implementation, the efficiency is increases, size of area decreases, complexity decreases and the computational cost is also reduced. In this paper we reduced the LUT by 11.34 % by using Gaussian filter, we can see in Table 1.

REFERENCES

1. M. C Hanumantharaju, M. ravishankar, and D.R Rameshbabu, “Design and FPGA Implementation of a 2-D Gaussian Surround Function with Reduced on-chip Memory Utilization” in proc. of the international conference on computer vision, pp 256-259, may 2013.
2. Frank Cabello, Julio Le6n, Yuzolana, t Rangel Arthur, “ Implementation of Fixes-point 2D Gaussian filter for Image Processing based on FPGA” in proc of IEEE Signal Processing algorithemic, architecture and application” PP 28-33, SEP 2015.
3. P. Burt, “Fast filter transform for image processing,” Computer graphics and image processing, vol. 16, no. 1, pp. 20–51, 1981.
4. P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” Communications, IEEE Transactions on, vol. 31, no. 4, pp. 532–540, 1983.
5. D. AIghurair and S. S. AI-Rawi, "Design of Sobel operator using Field Programmable Gate Arrays," in 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). IEEE, May 2013, pp.589-594.
6. V. Sriram and D. Kearney, "A FPGA Implementation of Variable Kernel Convolution," in Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007). University of South Australia: IEEE, Dec. 2007, pp. 105-110.

7. R. E. W. Rafael C. Gonzalez, Digital Image Processing using MATLAB, 2009, vol. 24, no.11.
8. D. G. Bailey, Design for Embedded Image Processing on FPGAs, 2011.
9. R. Sass and A. G. Schmidt, Embedded systems design with platformFPGAa ,2010.
10. P. Pong, FPGA proto typing by vhdl examples, 2008.
11. v. Pedroni, Digital electronics and design with VHDL, 2008.
12. C Hanumantharaju, M. Ravishankar, and D. R. RameshbabuDesign of Novel Algorithm and Architecture for Gaussian Based Color Image Enhancement System for Real Time Applications, In Proceedings of International Conference on Advances in Computing, Communication, and Control (ICAC3), Vol. 361, pp. 595-608, 2013.
13. A. Bhatia, W. Snyder, and G. Bilbro, "Stacked Integral Image," in Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010, pp. 1530–1535.
14. R. Deriche, "Using canny's criteria to derive a recursively implemented optimal edge detector," International journal of computer vision, vol. 1, no. 2, pp. 167–187, 1987.