A Novel Energy Efficient Gaussian Smoothing Filter for Image Processing

Dileep Singh Kushwah[1] , Mrigendra singh[2] and Vijay Kumar Magraiya[3]

[1]Research scholar, [2]Research scholar,[3]Assistant Professor,
Electronics and Comm. Engg. Deptt., SRCEM Banmore, Morena, India

***Abstract*—The VLSI design for today's application has become very complex and requires huge computational power to satisfy users demand. In order to utilize the design into the battery operated devices it must be energy efficient. So the energy efficiency is the prime concern for today's VLSI design. In this paper, we propose a novel energy efficient approximate Gaussian Smoothing Filter that provides better results over the conventional approach. The proposed design is implemented in Tanner EDA tool to obtain design metrics and modeled in MATLAB to get error metrics. Simulation results shows that proposed filter improves the area, delay and power by 33%, 14%, and 25% respectively. Moreover the design improves energy by 35% over the existing design at the cost of 6.68dB loss in PSNR.**

***Keywords*—Digital signal processing (DSP), Image processing, High-speed integrated circuits, VLSI, Low power design, Peak signal to noise ratio (PSNR), Mean squared error (MSE).**

I. INTRODUCTION

With the increasing improvement in the VLSI technology more and more functionality are being integrated on the chip to satisfy user requirements. This increased functionality on the chip results in degradation in the parameters such as area, power and delay. The portable multimedia devices which demands high energy efficiency restrict the use of image/video processing applications [1]. Thus, there is a high demand of the energy efficient design on the portable devices. The high power consumption within these image/video processing applications is due to large mathematical operation. Thus, the complexity of these designs can be reduced by reducing the number of operations. In image processing noise signals may introduce during the stages of compression, acquisition, processing, storages or even transmission. In addition to the noise at image capturing time, noise may occur during transmission and storage. To eliminate the effect of noise signals, we require filters in image/video applications. The noise reductions techniques generally involve local pixels averaging, and/or generating a blurred or smoothed image. Hence, the noise reduction process is sometime called as smoothing process. Smoothing filter is used for blurring [2] and for noise reduction. In order to remove small details from an image prior the object extraction, and bridging of small gaps in line or curves, blurring is used in preprocessing step of image processing. Further, the image quality metrics can be classified according to the availability of an original image over which the distorted image is compared. The simplest and most commonly used quality metric is the mean squared error (MSE), computed by averaging the squared intensity differences of distorted and reference image pixels. Moreover, peak signal-to-noise ratio (PSNR) is also found to the metrics used to evaluate the quality of the design.

In edge detection process, to simplify the analysis of images [3], Gaussian smoothing is done earlier to Laplacian to remove the effect of noise. Gaussian smoothing is a special case of weighted averaging smoothing [4], where the coefficients of the smoothing kernel are derived from a Gaussian distribution. It has been mainly adopted in image processing. The Gaussian filter shows an important role in digital image processing such as image blurring, image segmentation, and edge detection.

From the aforementioned analysis, we found that we should design A novel Energy Efficient Gaussian smoothing filter that can provide a large range of noise reduction ability into hardware to advance the range of the noise immunity of the hardware-based edge detection algorithm. This is only possible by applying approximation technique into the Gaussian filters.

II. BASIC OF GAUSSIAN SMOOTHING FILTER

The image smoothing can be established via different types of filters, the most commonly used in image processing is Gaussian smoothing filter that we will discussed in this paper. In the Edge detection techniques, we have adopted the 2D Gaussian for mainly three main important reasons.

The first reason is that when combined with a Laplacian operator, the 2D Gaussian filters that do not create fake edges as the scale increase. The second reason is that Gaussian smoothing provides the finest tradeoff between localization in both spatial and frequency domains. And finally third reason is Gaussian filters are the only rotationally invariant 2D filters that are separate in the horizontal and the vertical information, which makes the convolution in the spatial domain very efficient.

A 2D Gaussian function with zero mean and standard deviation σ is be described by [7]:

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

Where, the function ideally calculates to non-zero for all values of *x* and *y*.

Let the image with frequency response F(m,n) and Gaussian Kernel with its frequency domain impulse response is K(m,n) produces smooth image G(m,n) by multiplying K(m,n) and F(m,n)

$$G\ (m, n) = K\ (m, n)\ x\ F\ (m, n) \quad (2)$$

where G(m,n), K(m,n), and F(m,n) are respectively the frequency domain representations of g(*x*,*y*), k(*x*,*y*), and f(x,y). Similarly the smoothed image in the spatial domain using the convolution expression

$$g\ (x, y) = k\ (x, y)*f\ (x, y) \quad (3)$$

Impulse response of the 2D Gaussian smoothing filter g(x, y) must be approximated by a finite number of coefficients, generally known as convolution kernel, or mask for efficient implementation [7]. For the index (*x*, *y*) of the center element is set to (0, 0) of a 5×5 mask, and the value of x and y are varied from (-2, -2) to (2, 2). From the definition of Gaussian distribution, the summation of the complete kernel should equal to 1. Therefore, each element must be normalized with reference to this sum of all elements.

$$g_n(x ,y)=g(x ,y) / N \quad (4)$$

Where, $N= \sum_{x=-2}^{2}\sum_{y=-2}^{2} g(x\ ,y)$

The 5x5 kernel coefficient values shown in Fig. 1 which are derive from equations (1) and (4)by varying x and y from -2 to 2 with standard deviation (σ) =1. Further the coefficients are rounded to the 4th decimal point.

$$\begin{pmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{pmatrix}$$

Fig. 1 5x5 normalized Gaussian kernel coefficients with σ=1

III.PRINCIPLE OF FIXED-POINT OPERATION

The fixed-point data is represent in the (l, m) pattern where l shows the number of bits used, and m shows the location of the coefficient least significant bit. For example, the decimal equivalent of the coefficient X_3 X_2 X_1 X_0 in a (4,-8) format is

$$X_3 * 2^{-5}+ X_2 * 2^{-6}+ X_1 * 2^{-7}+ X_0 * 2^{-8}$$

The fixed point operation of FIR filters with power of two coefficients. For the Gaussian smoothing filer, each coefficient is round to the nearby value within the available bit-width budget. Due to the high complexity of implementing multiplier and dividers these are implemented by shift and add operations. Where, the multiplication and the division operation which are in powers of two are implemented by right shift and left shift operation.

The Gaussian smoothing kernel [7] coefficients of Fig. 1 are rounded to the (6,-8) data format and are shown in Fig. 2

$$= \frac{1}{2^8}\begin{pmatrix} 1 & 3 & 6 & 3 & 1 \\ 3 & 15 & 25 & 15 & 3 \\ 6 & 25 & 41 & 25 & 6 \\ 3 & 15 & 25 & 15 & 3 \\ 1 & 3 & 6 & 3 & 1 \end{pmatrix}$$

Fig. 2 Gaussian kernel with (6,-8) data format

Fig. 3 shows a Gaussian smoothing architecture of (6-8) data format where we can see that it requires more number of adders to implement the design. Further the design does not require any other module such as multiplier.

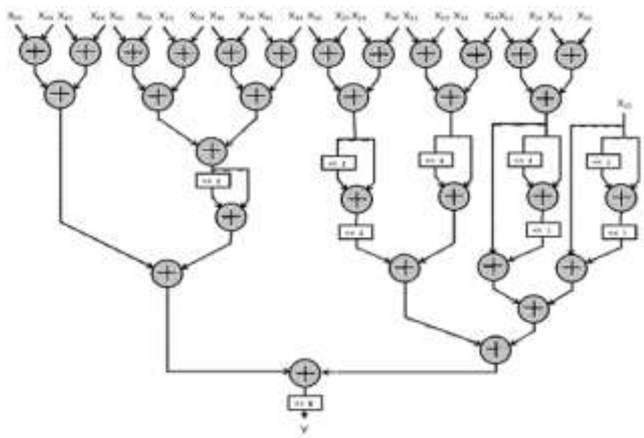

Fig. 3 Architecture of Kernel 5x5 with (6,-8) data format

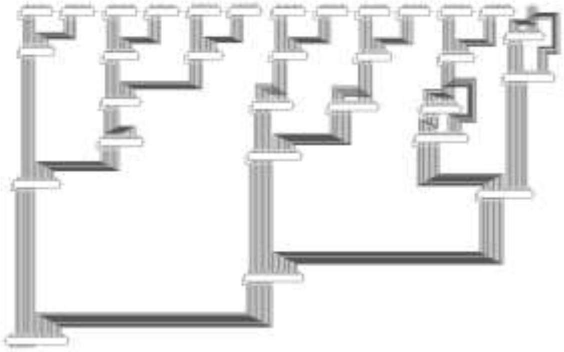

Fig.4 Schematic figure of 5x5 Kernel with (6,-8) data format

IV. PRAPOSED WORK

Previous work in Image/video processing Gaussian smoothing filter provides various limitation of:

1. High implementation complexity due to complex multiplicative terms.
2. High power consumption and low performance due to large complex and large multipliers.

With the aim of reduce these limitations and increase the energy efficiency, we represent a novel energy efficient Gaussian smoothing filter for image processing with a little adjust in kernel coefficient. We can design energy efficient Gaussian smoothing filter compared to conventional design kernel, with little loss in accuracy. It provides significantly improved power, delay and area parameters and more energy efficient.

In this paper, we represent a proposed a novel energy efficient Gaussian smoothing kernel of fig. 2.with a little change in Kernel coefficients. Fig. 5shows the proposed Energy Efficient Gaussian smoothing kernel

$$= \frac{1}{2^8} \begin{pmatrix} 0 & 4 & 8 & 4 & 0 \\ 4 & 16 & 24 & 16 & 4 \\ 8 & 24 & 40 & 24 & 8 \\ 4 & 16 & 24 & 16 & 4 \\ 0 & 4 & 8 & 4 & 0 \end{pmatrix}$$

Fig. 5 Proposed approximate 5x5 Kernel

The architecture of the proposed kernel is shown in fig. 6. It requires less hardware compared to the original design. This result less power is used and area is also reduced. So this design is much energy efficient than conventional design.

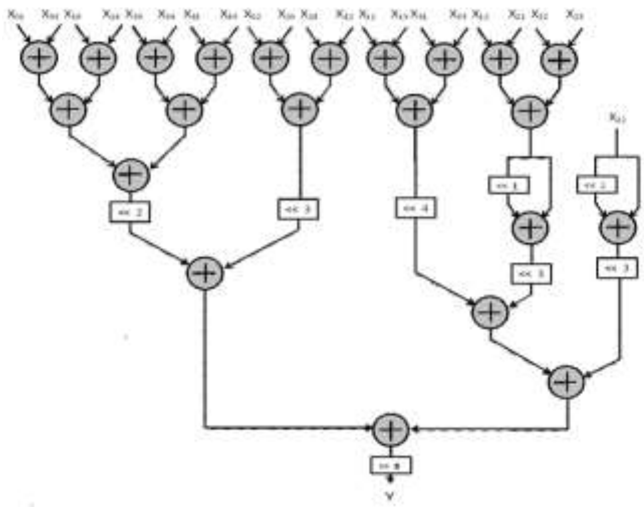

Fig. 6: Proposed GSF architecture of kernel in fig. 5

With the aim of evaluate all the design metrics we implemented all the above mentioned architectures for the Gaussian smoothing kernels, we utilize the tanner 14.1 EDA tool.

The schematic figure of full-adder with 28 transistors is shown in Fig. 6(a). We utilize this full adder to design different bit-width ripple carry adder (RCA) and it schematic figure is shown in Fig. 6(b). These adders of different bit-width are used to design the schematic architecture of the proposed filter as shown in Fig. 6(c).

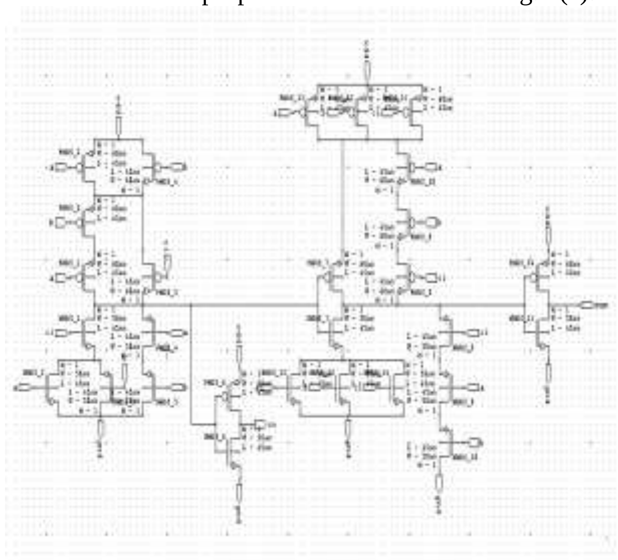

Fig. 6 (a) Schematic 28T figure of Full Adder

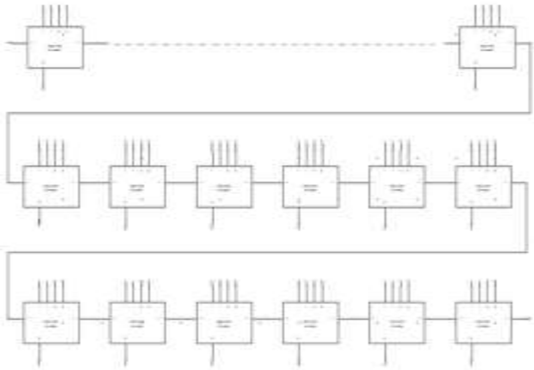

Fig. 6(b) Schematic figure of Ripple Carry Adder

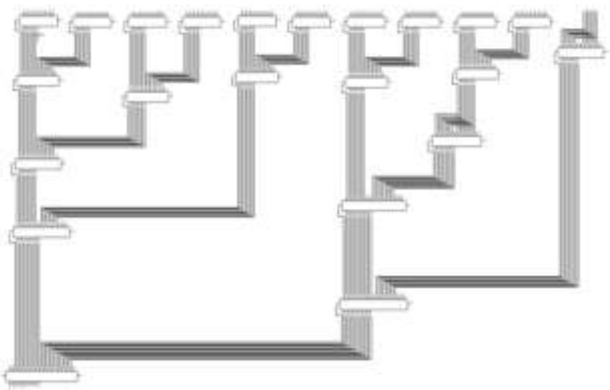

Fig.6(c) Schematic of proposed kernel 5 x5 fig. 5

Furthermore, all the designs are modelled on the MATLAB to evaluate the error metrics. Lena (512x512) noisy image is filter via filter reported in [7] and via proposed GSF.

V.EXPERIMENTAL RESULT & ANALYSIS

This section first introduces design metrics and then error metrics to evaluate the proposed design.

Design Metrics: In order to estimate the design metrics the proposed energy efficient Gaussian smoothing filter is implemented on the Tanner 14.1 and simulated with 45nm technology file. To have fair comparison transistor sizing are taken identical with same power supply for the proposed and reference design.

Table I: Comparison of design metrics

Design	Area (#MOSFET)	Power (µw)	Delay (ns)	Energy Efficiency (pJ)
GSF [7]	9128	224.1	0.116	0.026
Prop.	6076	167.47	0.10	0.017

It can be seen that proposed energy efficient Gaussian smoothing filter reduces the area, power and delay by 33%, 25%, and 14% respectively, over the GSF reported in the paper [7]. Moreover the design improves energy by 35% over the existing design.

Error Metrics: The error metrics after simulating noisy Lena image is shown in Table 2. Moreover, the noisy Lena image after smoothing via GSF reported in [7] and proposed energy efficient Gaussian smoothing filter is shown in Fig. 6. It can be seen that although the error metrics look poor over conventional but the noise in completely acceptable in the proposed design and the proposed Gaussian filter can be effectively applied in any image processing applications.

Table II: Error metrics comparison

Design	Mean Error (µ)	MSE	PSNR (dB)	Std. dev. (σ)
GSF [7]	2.3631	21.56	80.11	4.6442
Prop.	4.9743	42.05	73.43	6.4848

(a) **(b)**

(c) **(d)**

Fig. 7 Lena (512x512) images (a) original image, (b) noisy image, (c) filter by GSF [7], and (d) filter by proposed energy efficient GSF.

VII. CONCLUSION

In this paper, we proposed a novel Energy Efficient Gaussian smoothing filter that provides significant reduction in power, delay and area with very small loss in accuracy. Moreover the design improves energy by 35% over the existing design. The novel Energy Efficient Gaussian smoothing filter provides the high speed by the approximating the kernel coefficient. Comparisons with conventional Gaussian smoothing filters showed that the proposed novel Energy Efficient Gaussian smoothing filter performed better than the all conventional Gaussian smoothing filters in power consumption, speed performance and energy efficient. Novel Energy Efficient Gaussian smoothing filter can be utilized in all those applications where there is no accurate requirement of accuracy or where ultra-low power and high-speed are more important than accuracy. The proposed design is best suitable in low power battery operated devices such as mobile and other gadgets.

REFERENCES

[1] A. V. S. S. Prasad, J. Mathews, and N. Naganathan, “Low-power design strategies for mobile computing, in *VLSI Design, 2006 19th International Conference on*, pp. 2, Jan 2006.

[2] T. Popkin, A. Cavallaro, and D. Hands, “Accurate and efficient method for smoothly space-variant gaussian blurring,” *Image Processing, IEEE Transactions on*, vol. 19, pp. 1362–1370, May 2010.

[3] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679–698, Nov 1986

[4] Pei-Yung Hsiaol, Chia-Hsiung Chen2, Shin-Shian Chou', Le-Tien Li', and Sao-Jie Chen2, “A Parameterizable Digital-Approximated 2D Gaussian Smoothing Filter for Edge Detection in Noisy Image” IEEE trans. 2006.

[5] Pei-Yung Hsiao1, Shin-Shian Chou2, and Feng-Cheng Huang1, “Generic 2-D Gaussian Smoothing Filter for Noisy Image Processing” IEEE trans. 2007.

[6] Firas Hassan, Sami Khorbotly, “Recursive Implementation of Exponential Linear Phase FIR Filters” IEEE trans. 2011.

[7] Firas Hassan, Sami Khorbotly, “A Modified Approximation of 2D Gaussian Smoothing Filters for Fixed-Point Platforms, “ IEEE trans. 2011