A Review Paper on Denosing Filter using 2D Gaussian Smooth Filter for Multimedia Application

Mithlesh Kumar Kaushik  
Electronics & Communication  
Dr. C.V. Raman Institute of science & Technology Bilaspur, cg India

Mrs. Rashmi Kasyap  
Electronics & Communication  
Dr. C.V. Raman Institute of science & Technology Bilaspur, cg India

Abstract—According To current technology there is lots of future scope in area of internet of things, Video/audio processing etc. For these are there is need of lots of sensor and multimedia design which is use full in these area. We also know in present era every system work on battery power only. So there is need of fast and energy aware system which will reduce the battery consumption issue and run system for long time. As we know in present stage ever one doing real time image/video transmission. Due to real time may be some time there is image quality will decrease so for improvement of those real time image there is need of De-noising approach which is well known as Smooth filter. In this paper basically we did a comparative study about the existing different types of De-noising filter.

Index Terms—Approximate design, Gaussian Smoothing Filter, Error Tolerant Applications, Energy-efficiency, Edge-detection.

I. INTRODUCTION

The present mobile and portable multimedia devices are suffering from energy crises due to exponentially increased functionality and complexity [1]. These devices incorporate image processing as their core application which suffers from noise while compression/transmission, and degrades the image quality. To reduce this noise, smoothing filters with characteristic such as averaging, median, mean and Gaussian etc. are employed. The commonly used 2D Gaussian Smooth Filter (2D−GSF) does not generate false edge on increasing scale and provides good trade–off between localization in spatial and frequency domains [2]. The well−known edge detection algorithms Canny [3] and Marr Hildreth [4] employ 2D−GSF. The 2D−GSF is also useful in many other applications such as texture segmentation [5], tone mapping of high dynamic range images [6], image blurring [7], and image mosaicing [8]. Existing approaches of 2D−GSF do not provide energy−efficient designs for portable devices, where energy−efficiency is crucial over quality. In order to overcome this limitation, approximate computation techniques can be exploited to achieve energy efficiency for these applications.

The approximate designs produce results with acceptable quality and offer power reductions with performance improvements in return [9]. These designs exploit error re−saliency/tolerance of the application to improve speed, power and area metrics. The confined human perception and ability to extract information from small noisy image allows small amount of error while processing image/video [10]. It allows approximate/imprecise designs for image processing applications to achieve better design metrics with acceptable quality. Considerable efforts in past few years demonstrate an exploration of approximate adders [11], multipliers [12], squarer [13] and DCT architectures [14]. Very few efforts in designing approximate 2D−GSFs are reported in the literature. Hsiao in [15] proposed an architecture of 5×5 window based 2D−GSF which uses approximate algorithm in power of two with small change in standard deviation (σ). The major drawback of the design is its large area requirement and uneven accuracy. The Gaussian kernel contains floating point coefficients that increase its implementation area and power consumption. Khorbotly et al. in [16] use fixed point data and rounded−off filter coefficients to reduce the computation time, but area inefficiency is still a major concern. To overcome this, we propose an area and energy efficient approximate 2D Gaussian smoothing filter with different kernel sizes. Our main contributions are:

- We propose energy−efficient architecture of 2D-GSF that significantly reduces implementation complexity.
- A new concept of nearest pixel approximation is introduced that reduces number of processing elements.
- We also derive an approximate 2D Gaussian kernel matrix with less number of coefficients.
- The proposed approximate filter is implemented and evaluated for edge detection application.
- We achieve 72%, 79% and 76% reduction in area, power and delay, respectively with acceptable (PSNR = 0.4dB) quality loss. The rest of the paper is organized as follows. Necessary background and underlying principle of 2D−GSF is given in Section II whereas, Section III describes proposed methodology. Section IV demonstrates architecture of the proposed 2D−GSF.
II. LITERATURE SURVEY

The fields of image processing and computer vision are continuously gaining increased attention in applications including robotics, automation, quality control, and security systems. Among the many image processing procedures, edge detection is seen by many as the first essential step in any type of image analysis. It is used to separate the image into object(s) and background. The performance of an edge detection operator is defined as its ability to locate, in noisy data, an edge that is as close as possible to its true position in the image.

EFFECT OF SMOOTH FILTER ON IMAGE:

![Original Image](image1) ![Box-filtered image](image2) ![Gaussian-filtered image](image3)

Kernels used for blurring:
(Note that the values shown have been scaled up to integers for clarity)

**2D GAUSSIAN SMOOTH FILTER:**

2GSF is one of the most common filter which is used in many image processing applications. This filter is based on a fixed value of standard deviation. The equation of a Gaussian function in 2D with a standard deviation can be described by:

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

Where x & y is distance from origin of horizontal & vertical axis. When this formula is applied on 2D image, it produces a surface whose contours are concentric circles with a Gaussian distribution from the centre point. When a noisy image t is applied to gaussian filter with impulse response of g, so spatial domain of smooth image k is calculated by using of convolution.

$$k(x, y) = t(x, y) * g(x, y)$$

Similarly, frequency domain of smooth image k is calculated by using of below expression,

$$K(a, b) = T(a, b) \times G(a, b)$$

Frequency domain is generated by spatial domain, which are k(x,y), t(x,y) and g(x,y). Here K (a,b), (a,b) and G(a,b) are represent frequency domain form. In equation (1) will decide amount of smoothness. For more effective smoothing large value of σ& large kernel is required for accurate representation of a function. In this paper we have proposed gaussian smooth filter of 3X3 and 5X5 kernel for σ=1, which is calculated by equation (1) for given (x,y) values. For calculation of 3 X3 kernel, value of x & y lies between -1 to 1. Coefficients of a 3X3 gaussian kernel for σ=1 are:

$$\begin{bmatrix}
0.0751 & 0.1238 & 0.0751 \\
0.1238 & 0.2042 & 0.1238 \\
0.0751 & 0.1238 & 0.0751 
\end{bmatrix}$$

For calculation of 5 X5 kernel, value of x & y lies between -2 to 2. Coefficients of a 5X5 gaussian kernel for σ = 1 are:

$$\begin{bmatrix}
0.0751 & 0.1238 & 0.0751 & 0.1238 & 0.0751 \\
0.1238 & 0.2042 & 0.1238 & 0.2042 & 0.1238 \\
0.0751 & 0.1238 & 0.0751 & 0.1238 & 0.0751 \\
0.1238 & 0.2042 & 0.1238 & 0.2042 & 0.1238 \\
0.0751 & 0.1238 & 0.0751 & 0.1238 & 0.0751 
\end{bmatrix}$$
2D gaussian smooth filter is basically based on gaussian kernel function, having a floating point coefficient. In terms of hardware complexity, floating point design requires large hardware unit and huge amount of energy. Existing approaches will make injustice with ASAP (Speed, Power, Area, Accuracy) metrics, while operating on portable devices. In order to justify ASAP metrics there is need of approximate design.

**Approximation for Error Tolerant Applications:**

The approximate designs produce almost-correct results, and offer power reductions with performance improvements in return. This design exploits a tradeoff of accuracy in computation versus speed, power and area. Explained with an example, let there be two number X=223 and Y=224. Its accurate and approximate multiplication [8] will results to Z = 49952 and Z = 46847. The total and percent error difference is 3105 and 6.21% respectively. As per [7], the human eye can tolerate an error upto 10%. So we can easily apply approximation on image processing system. Through this small error there is tremendous saving in hardware complexity.

**Working of Approximate Multiplier [8]:**

The multiplication process begins at the point where the bits split and move simultaneously towards the two opposite directions till all bits are taken care of. In the example of Fig. 1, the two 12-bit input operands, the multiplicand "10110011011" (2971) and the multiplier "010011001001" (1225), are divided into two equal-sized parts, and each of which contains 6 input bits. As for the lower order bits of the input operands (non multiplication part), a special mechanism is applied – no partial product will be generated and the carry propagation path has been removed. Every bit position from left to right (MSB to LSB) of the non-multiplication part is checked and if either or both of the two operand bits are "1". The checking process is brought to an end and from that bit onwards, all the bit positions are set to "1". In the event that both operand bits are "0", the corresponding product bit is set to "0". In this way, the overall error generated due to the elimination of partial-products can be minimized. In the example, at the sixth position, the two input bits are both equal to "0". Hence, the corresponding result bit is set to "0". At the fifth LSB bit (2nd position from the starting point), as the multiplicand bit is "1", the corresponding result bit is set to "1" and all the remaining and all the remaining resultant bits to its right are also set to "1". For the higher order bits of the input operands that fall into the multiplication part, the operation is conducted as per in normal multiplication operation, from right to left (LSB to MSB). Its circuit is hence constructed in the conventional way. We retained the conventional topology here since the higher order bits have greater weightage than the lower order bits. By eliminating the partial products and the carry propagation path in the non-multiplication part (LSBs) and performing the multiplication of the MSBs simultaneously, the overall delay time is greatly reduced.
WORKING OF APPROXIMATE ADDER [9]:

For conventional digital circuit design, the power and speed is the component to trade off each other. Usually, if the system requires a high speed performance, high power consumption is needed; vice versa, if the system consumes very low power, the speed will be slow. For Error-Tolerant adder, a new component to trade-off is introduced: the accuracy. This means if the system can accept some error, both power and speed can be improved. When the accuracy of a circuit is brought into the design, the two dimensional trade-off between power and speed becomes three dimensional, i.e. power-speed-accuracy. Adding new components provides the system designers with more flexibility: if low-power and high speed are desired, the accuracy can be used as a trade-off.

III. PROBLEM IDENTIFICATION

As we know in present era everyone need fast & low power system with good quality at real-time. But according to previous existing approach there is lots of issue with time complexity, power, area. All previous accurate algorithms are good in quality level but hardware complexity & latency issue is too much. So for reduction of those issue approximate 2D Gaussian smooth filter are come to picture.
So According to those previous existing approximate 2D Gaussian smooth filter algorithms are reduce some problem like justification with ASAP metrics. But image visual quality is not up to the mark. [15], [16] are able to reduce power, area and speed metrics but for more noisy image they are not suitable to make justice with quality.

So these all are the previous existing problems:
1. Having a issue on Latency Complexity
2. Having a issue on Area Complexity
3. Having a issue on Power Complexity
4. Quality Complexity

IV. FUTURE OBJECTIVE

In future this proposed method will be used in wide areas like Multimedia Applications, Face Recognition, Edge detection etc. In this research area, still there is lots of future work is require. The main focus is on performance and accuracy, but we do provide some numbers for the arithmetic units relating to energy and power. This is to provide an estimate of the amount of energy and power consumed by the units we choose to implement. The priorities of the future research objectives are, in order of importance, are:
1) High performance
2) Low complexity
3) power/Area /Accuracy

V. CONCLUSION

According to our previous research we got there is lots of issues with the multimedia application in terms of quality, time and energy complexity. As we already know there is lots of scope of De-noising filter, I present era everyone need HD quality images/videos so for those applications there is need of De-noising filter. According to previous research there is lots of problems are there which we can still resolve. So in this paper basically we did the comparative analysis between existing approaches of de-noising filter.

REFERENCES


