ISSN 23499842(Online), Volume 1, Special Issue 2(ICITET 15), March 2015
International Journal of Innovative Trends and Emerging Technologies

AN FPGA IMPLEMENTATION OF A SYNCHRONIZED BI-LATERAL FILTER FOR IMAGE PROCESSING

MANOJ KUMAR.D[1], MRS. KASTHURI BHA .J.K[2]
[1]M.Tech VLSI Design, [2] Asst Professor,
[1,2] SRM University

Abstract*— Images are often corrupted by noise in the procedures of image acquisition and transmission. In this paper, we propose an efficient de noising scheme and its VLSI architecture for the removal of random-valued noise. To achieve the goal of low cost, a low-complexity VLSI architecture is proposed. In this paper, a detailed description of a synchronous field-programmable gate array implementation of a bilateral filter for image processing is given. The bilateral filter is chosen to reduce noise while preserving details. Edges are basic properties of images. The distinctive feature of our design concept consists of changing the clock domain in a manner that kernel-based processing is possible, which means the processing of the entire filter window at one pixel clock cycle. The complexity of the design is widely reduced by symmetrical block based approach. Combining these features, the bilateral filter is implemented as a highly parallelized pipeline structure with very economical and effective utilization of dedicated resources. Due to the involvement of booth multiplication algorithm the architecture can be implemented with low effort and speed will be increased considerably without compromising complexity.

Index Terms—Bilateral filter, field-programmable gate array (FPGA), image processing, noise reduction, real-time processing.

I.INTRODUCTION

Bilateral filtering has gained great popularity in image processing due to its capability of reducing noise while preserving the structural information of an image. The bilateral filter [consists of two components. The detail-preserving property of the filter is mainly caused by the nonlinear filter component also called photometric filter. It selects the pixels of similar intensity which are averaged by the linear component afterward. Very often, the linear component is formulated as a low-pass filter. The amount of noise reduction via selective averaging and the amount of the blurring via low-pass filtering are both adjusted by two parameters. The understanding of these parameters is very intuitive, which leverages the bilateral filter to an almost all-purpose solution in image processing. The bilateral filter is applied for noise reduction in a method for local tone mapping which maps high dynamic range image to low dynamic range image. Recently, bilateral filtering has gained a high awareness level in medical image processing and non-destructive testing. This means that the noise filtering takes place prior to computing the reconstructed volume. It has been concluded that noise reduction of this kind can be translated into a dose reduction in X-ray computed tomography. Considering industrial applications, the dose reduction permits the reduction of the scanning time and thus allows a higher throughput of test items. As the reduction of the exposure time due to filtering is feasible, we are interested in a real-time filtering of projections. Moreover, the filter is not supposed to reduce the spatial resolution of projections to maintain the visibility of defects in a reconstruction. Since we achieve very satisfying results considering detail preservation with our field-programmable gate array (FPGA) implementation we intend to give a deeper insight in our work. The major contribution of this paper is the detailed description of a novel FPGA design architecture of the bilateral filter on register-transfer level (RTL). This abstraction level is chosen for the possibility of direct specification of the clocking scheme.

II. BILATERAL FILTER ARCHITECTURE

. The architecture consists of three main blocks, the register matrix, photometric filter and the geometric filter. The input image is given as a serial input to the register matrix where it is formed into groups. The photometric filter acts as the range filter while the geometric filter performs filtering in the domain.

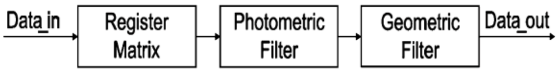


Fig 1. Bilateral Filter Architecture

(A) Register Matrix

The Register Matrix is used as the input mechanism for the filter. For the design description, a window size of 5 × 5 is chosen. This window size is the trade-off between high noise reduction and low blurring effect. Consequently, the demand arising from this filtering technique is that at least five lines have to be stored for the period of time during which a line is filtered. As an external image buffer is undesired because of the additional expenses of resources due to the memory controller and because of the additional latency due to the memory accesses, the five input lines are stored in the line storages which are implemented as block RAMs for data with N bits. The five input lines are called image rows or rows in the following. These five rows include the row to be filtered, two foregoing rows, and two succeeding rows. This arrangement is depicted in the fig. The pixel being filtered is marked by “mid_pix.” This pixel and its neighbourhood in the solid box represent the kernel of the bilateral filter. As the input data are read into the register matrix pixel by pixel, the content of the line storages and of the filter kernel is shifted by one pixel at each clock event. The output of the register matrix is sorted into groups, in this case into six groups, and fed into the photometric filter component. The sorting is done by means of multiplexing the pixels in the manner shown in Fig. 2. The counter on the top of Fig. 2 generates the select signal and thus controls the readout of the register matrix. The counter is first enabled after the whole register matrix is filled. The pixels in each group are processed in parallel while each group is pipelined through to the register matrix output stage. The pixel in the centre of the filter window is not a part of any group and is forwarded to a latch belonging to the input stage of the photometric filter component. The sorting of the pixels into groups and the quadruplicating of the pixel clock are the key to the presented synchronous FPGA design concept using a parallelized pipeline architecture. The image is split up and fed into the register matrix as shown below in that particular order.

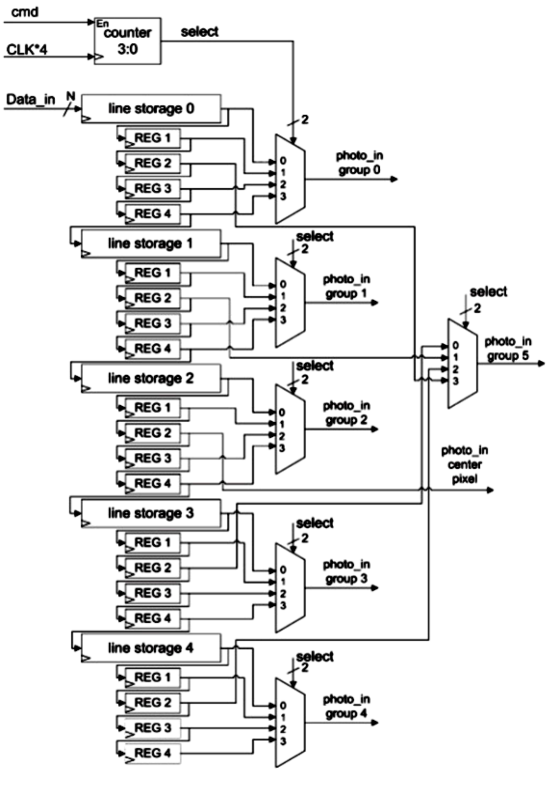


Fig 2. Register Matrix

(B) Photometric Filter.

After the register matrix has been filled, the grouped image data are provided to the photometric filter component which is pictured in Fig.3. At the output of the photometric filter, the weighted pixels appear, still sorted into groups, accompanied by the “weighted mid_pix.” Additionally, the photometric coefficients have to be forwarded for the required normalization at the last stage of the filtering. For this reason, the output of the photometric filter consists of the following:

1) Weighted pixels sorted into groups 0...5.

2) The weighted pixel being filtered, marked by “mid_pix”.

3) Photometric coefficients corresponding to groups 0...5.

In further stages of the design, the weighted pixel values, i.e., the outputs of the multipliers, are named by their groups 0...5The pixel in the centre of the filter window has to be available during the calculation of the required 24 pixel weights. Latching the centre pixel allows the computation of the grey value differences between the centre pixel and the remaining pixels inside of the filter window. Each group contains four pixels. A separate pipeline belonging to each group makes it possible to process the entire neighbourhood of “mid_pix” at one pixel clock signal. All six pipelines are designed identically. At the first internal clock event t0, the first pixels of each group are provided to the respective pipeline. At the second internal clock t1, the second pixels of each group enter the component. This organization of groups allows the processing of the whole filter window in four internal clock cycles corresponding to one pixel cycle. Through the following, we use registers to keep our design synchronous. Thus, it makes any delay control inside of our architecture redundant.

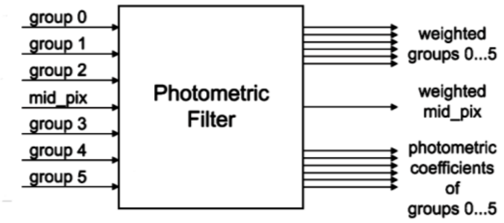


Fig 3. Photometric Filter.

To avoid the calculation of the expensive exponential, all possible values of the photometric function are recalculated and stored in the lookup table (LUT). The absolute difference of the grey values itself is directly interpreted as the address of the corresponding weight coefficient in the LUT in the standard system there are only 188 coefficients to be stored. The maximum number of coefficients that can be stored is given my 2P-1. The coefficients are stored in the LUT of each pipeline in the initialization phase of the filtering. The result of the disjunction selects the coefficient address. If the grey value difference is greater than the limit, the weight coefficient is set to zero which is stored at the address 2P −1. In the opposite case, the corresponding coefficient is read out of the LUT. This coefficient may also be zero as the number of coefficients is extended to 2P −1. During the readout of the coefficient, the related grey value is registered for synchronicity. At the next internal clock event, the grey values of each group are multiplied by the corresponding coefficients. The pixel in the centre of the filter window does not belong to any group and is processed separately. The output of the photometric component is forwarded to the Geometric filter for the next stage of filtering.

(C) Geometric Filter

The geometric filter component is designed taking advantage of its separability and its symmetry. Because of this property the geometric filter is split into the vertical and horizontal parts. Therefore, 2-D filtering is replaced by successive 1-D filtering in vertical and horizontal directions. This solution is preferred in the design of the geometric filter because 1-D filtering can be implemented more efficiently. Due to the symmetry of the weight coefficients of the geometric component, the order of multiplication and addition is swapped in both filter parts. At first, the weighted grey values which are located at the same distance from the centered pixel in the filter window are summed up. For the simplicity of the design, it makes sense to assemble the pixels into equally large groups. Smaller groups allow for better handling of the design. For this reason, the pixels are divided into groups of four with regard to the subsequent processing explained in the following sections.

1) Vertical Component Part

The first stage of the geometric component is the vertical part which is pictured in the Fig. It can be seen that the pixels of the first column numbered 1, 2, 3, 4, 5 and the first pixel of the middle column numbered 11 enter the vertical component part simultaneously. For the corresponding photometric coefficients, the same order of processing is valid. The geometrically symmetrical pixels are cumulated at first and then multiplied by the geometric weight coefficient. After the accumulation of the pixels according to their symmetry, the sum is multiplied by the corresponding coefficient. The horizontal processing is done in the same way. The geometric coefficients are calculated in advance and stored in a block RAM. The registers in this part of the design are used to delay weighted data to maintain synchronicity. After the multiplication, the weighted values are summed up by the adder tree to one value at each internal clock event. The centered pixel is weighted and delayed so that this pixel and the remaining pixels in the centered column can be fed to the input of the adder tree simultaneously. The remaining pixels enter the dedicated processing path one by one. They were multiplexed in the register matrix in the way that they can be combined pairwise and multiplied by the same coefficient in the geometric Filter.

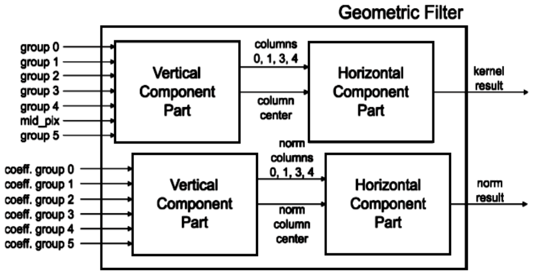


Fig 4. Geometric Filter

2) Horizontal Component Part:

After processing in the vertical dimension, the filter window is reduced to one row, and its elements are computed at one internal clock event each. In order to be able to reuse the symmetrical design, the values of the filtered columns 0, 1, 3, 4 are stored in the shift registers according to the order of their reception. The filtered photometrical coefficients are stored in the same way. The pixels undergo a change in the domain.The division is implemented through a shift operation. The remaining processing is similar to the processing described in the previous paragraph. The geometrically symmetrical pixels are cumulated at first and multiplied afterward by the geometric weight coefficient. For the geometric filtering in the horizontal direction, the same geo- metric coefficients are used as for the vertical filtering. The final division by the normalized one is performed in the next stage.

D) Normalization

At the final stage, the kernel result has to be normalized by the norm result as shown in Fig.5. After the final accumulation of these values, they are both divided by the normalized one again. In this manner, the word lengths of the weighted grey values and of the norm are both (W −1) bits shorter. Finally, after the division, N bits of the final result are forwarded to the output of the bilateral filter.

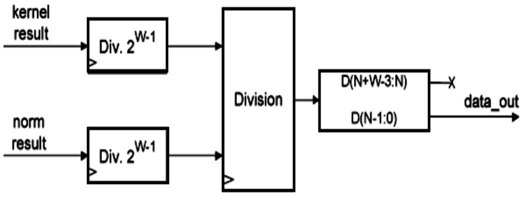


Fig 5. Normalization.

E) Power Reduction Using Multipliers.

Multipliers form the basic building blocks of a Filter. The standard Bilateral Filtering architecture proposed by the authors use a conventional array multiplier which is one of the most basic forms of multiplier. The main disadvantage of the array multiplier is the worst-case delay of the multiplier proportional to the width of the multiplier.. Also the Power consumption of the array multiplier is high compared to other multipliers. Therefore it is imperative to decrease the power consumption of the multipliers and thereby decrease the power consumption of the entire system. For this purpose we use Booth multipliers as an effective replacement for array multipliers. The Power Analysis and Timing Analysis of the architecture is carried out by Cadence RTL compiler and the results are tabulated below.

	Power (mW)	Delay (ns)
Standard Architecture using Array multipliers	35.5	12.3
Modified Architecture using Booth multipliers	28.7	10.8

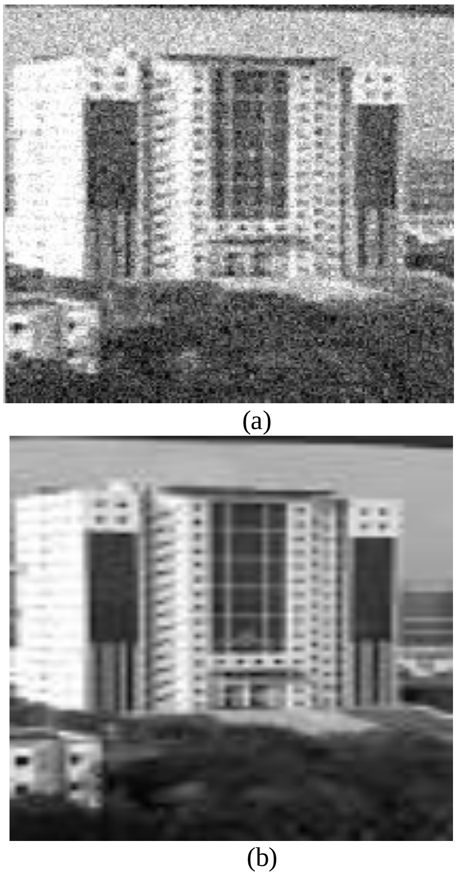
Table 1. Power Consumption Comparison

III. Results

IMAGE QUALITY ASSESSMENT

To evaluate the performance of the noise reduction and the accuracy of the detail preservation, criteria for the image quality assessment are required. The criteria chosen in this work is PSNR and MSSI. The Input noisy image used is shown in the fig (a). The input image is converted to hexadecimal format and fed into the register matrix pixel by pixel in which it is formed into groups. The image then undergoes range filtering and domain filtering in the photometric and geometric filters respectively. The output de-noised image is shown in (b). It is observed that the output image has a blurring effect due to the effect of filtering of high frequency component. The Mean structural Similarity Index (MSSI) is used to determine the structural similarity of the input and output images. The process can be extended to other images as well. The efficiency of the filtering process can also be improved by using Local Binary Pattern (LBP) .

Gaussian Noise	PSNR (dB)	MSSI
10 dB	52.7758	0.98
30 dB	50.7701	0.98
50 dB	48.7695	0.98



(a)

(b)

FIG 6: (A) INPUT NOISY IMAGE, (B) DE-NOISED IMAGE

IV. CONCLUSION

In this project we propose an effective bilateral filtering architecture. The standard architecture suffers from drawbacks such as large number of logic cells, high power consumption etc. We modify this standard architecture by replacing the standard multipliers in the bilateral filter with energy efficient and high speed Booth multipliers. The power analysis is carried out using Cadence Encounter RTL compiler. The two architectures are compared and the results are tabulated. The individual blocks of the filter are designed using Verilog. A noisy image is used as an input and the filtering process is applied. The parameters are calculated and tabulated.

V. REFERENCES

[1] C. Tomasi and P. Manduchi, “Bilateral filtering for gray and color im- ages,” in Proc. IEEE ICCV, 1998, pp. 839–846.
[2] B. Zhang and J. P. Allebach, “Adaptive bilateral filter for sharpness en- hancement and noise removal,” IEEE Trans. Image Process., vol. 17, no. 5, pp. 664–678, May 2008.
[3] B. Yan and A.-D. Saleh, “Structure enhancing bilateral filtering of images,” in Proc. IEEE PCSPA, 2010, pp. 614–617.
[4] M. de-Frutos-López, H. Medina-Chanca, S. Sanz-Rodríguez, C. Peláez- Moreno, and F. Díaz-de-María, “Perceptually-aware bilateral filter for quality improvement in low bit rate video coding,” in Proc. IEEE PCS, 2012, pp. 477–480.
[9] A. Gabiger-Rose, R. Rose, M. Kube, P. Schmitt, and R. Weigel, “Noise adaptive bilateral filtering of projections for computed tomography,” in Proc. 11th Int. Meet. Fully Three-Dimens. Image Reconstruction Radiol. Nucl. Med., 2011, pp. 306–309.
[5] J. Won Lee, R.-H. Park, and S. Chang, “Noise reduction and adaptive contrast enhancement for local tone mapping,” IEEE Trans. Consum. Electron., vol. 58, no. 2, pp. 578–586, May 2012.