

A Novel Architecture of Bilateral Filter for Image Denoising

Manju.C.Parvathy

Department of Electronics and Communication
Jawaharlal college of Engineering and Technology
Palakkad, India

Sreejith. S. Nair

Department of Electronics and Communication
Jawaharlal college of Engineering and Technology
Palakkad, India

Abstract—In this paper, a novel architecture for bilateral filter is described. The bilateral filter is an edge preserving filter that takes both the geometric closeness and photometric similarity into account. The distinctive feature of our design concept is the pipelining of separable modules. The input data is arranged into groups and internal clocks are provided for this kernel-based design. For acceleration of the design concept the separability and symmetry of the geometric filter component is utilized, also reducing the complexity of the design

Keywords—Bilateral filtering; image denoising; ; pipelining

I. INTRODUCTION

Images are often corrupted by random variations in intensity, illumination or poor contrast and can't be used directly. The process of removing noise from an image is called image denoising. Noise suppression in images not only improves its visual quality but also improves the compressibility of the image. There are many types of image denoising techniques which includes the employment of various filters. But most of these filters do not preserve the

details at the edges. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The detail preservation of edges are particularly very important in medical imaging. Bilateral filters are one of the most popular edge preserving filter. Bilateral filter was coined by Tomasi and Manduchi [1] in 1998. In bilateral filtering, how much a pixel in the vicinity of the current pixel (center pixel) can influence the filter output depends on its location and its value. If a pixel is near the center pixel, a high weight is assigned to it; otherwise a low weight is assigned. If a pixel has a value that is quite different from the center pixel, then a low weight is assigned. The final weight a pixel has is the product of these two weights. Computation of bilateral filter is performed based on sliding window. A pixel an its neighbourhood is taken. As we slide from one pixel to another, its spatial neighbourhood also changes in the same direction.

Bilateral filtering technique presents excellent filtered results but its application is limite by its speed. For each pixel in the raster scan order, neighbor pixels within a window around it are taken and used to compute the filter output; the window is moved right by one pixel and so on . Due to the large amount of computations in such an approach, a real-time implementation of the filter, is extremely challenging. Several accelerations have been proposed for the bilateral filter. In this paper we are describing the FPGA design concept of bilateral

filter. Filtering is based on the grey scale images and the size of filter window is 5×5 . This filter is designed for Gaussian noise and achieves convincing results in simulations.

II. BILATERAL FILTER

The denotation bilateral filtering suggests the combination of domain and range filtering. The nearby pixel values is averaged by the domain filter and thus acts as a low-pass filter. The range filter averages only the similar pixel values regardless of their position in the filter window and plays an important part in edge preserving. In case the value of a pixel in the filter window diverges from the value of the pixel in the center of the window by a certain amount, the pixel is skipped.

According to the geometric closeness an photometric similarity two weights are assigned for each pixel. The filtered pixel is given by the expression(1),

$$I'(u) = \frac{\sum_c e^{-\frac{\|u-p\|^2}{2\sigma_c^2}} \sum_s e^{-\frac{|I(u)-I(p)|^2}{2\sigma_s^2}} I(p)}{\sum_{p \in N(u)} e^{-\frac{\|u-p\|^2}{2\sigma_c^2}} e^{-\frac{|I(u)-I(p)|^2}{2\sigma_s^2}}} \quad (1)$$

Where, u and p represents the co-ordinates of centre pixel and spatial nieghbourhood respectively. $I(u)$ and $I(p)$ represents the grey values of centre pixel and spatial neighbourhood. σ_s and σ_c denotes the spatial and range parameters respectively. The following expressions (2) and (3) represents the photometric and geometric components respectively

$$s(I(u), I(p)) = \exp\left(-\frac{1}{2}\left(\frac{\|I(u) - I(p)\|^2}{\sigma_s^2}\right)\right) \quad (2)$$

$$c(u, p) = \exp\left(-\frac{1}{2}\left(\frac{\|u - p\|^2}{\sigma_c^2}\right)\right) \quad (3)$$

Weighting is followed by normalization an is given by expression (4)

$$k(u) = \sum_{m \in F} s(I(u), I(m)) \cdot c(u, m) \quad (4)$$

The photometric component compares the gray value of the

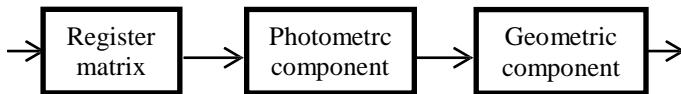


Fig.1 functional units of the bilateral filter

centered pixel with the gray values of the spatial neighborhood and computes the corresponding weight coefficients depending on the factor σ_s . The geometric component acts as a low pass filter that weights the pixels according to geometric closeness. The weights are reciprocally proportional to the distance between the centre pixel and spatial neighbourhood. As σ_s is increased further, Gaussian blur increases and as σ_c increases, smoothening increases.

III. DESIGN CONCEPT

The image data, as well as all constants and coefficients used in the following design concept, are integer numbers. With the aid of the presented design concept, the bilateral filter can be realized as a highly parallelized pipeline structure giving great importance to the effective resource utilization. In this paper, the data paths are detailed. The description of the control signals is not addressed here.

This design is based on 5×5 window size. The design concept is subdivided into three subsections. Fig. 1 presents these units and their order in the concept. First, the pixel values that are extracted should be sorted to different groups. Then the values of geometric and photometric component should be calculated and multiplied with the pixel value. This is followed by normalization. So the first step is to sort the pixel value into groups and store each value in a register. This is done in register matrix. It is followed by photometric and geometric component.

A. Register matrix

Basically register matrix does the grouping of rows so that elements can be used for pipelining in the next stage. Computing of mid pixel has to be done separately so that mid pixel is also separated. The values are computed using the MATLAB code. Fig.2 shows the design concept of register matrix. First a pixel is extracted from the picture and value are transformed to a 5×5 matrix. Each row is sorted from those values and it is saved in the workspace. The pixel weights for the photometric component has to be calculated for each of the pixel and the number of weights is dependent on the filter window size. Here, the size of the window is 5×5 . So there is a centre pixel and 24 spatial neighbourhood. Therefore a total number of 24 weights are to be calculated. Simulink, together with XSG and Xilinx ISE design software packages, is used for the filter implementation. Using xilinx system generator, the FPGA design is carried out using combination of simulink and xilinx blocks. The gray scale image data is read sequentially from MATLAB workspace. Image rows are called from workspace. Pixel values of each row are stored in a single port RAM. At every pixel clock the filter window is shifted one pixel. The output is sorted into 6 groups. Grouping of rows is done so that elements can be used for pipelining in the next stage. The counter controls the read out of the register matrix.

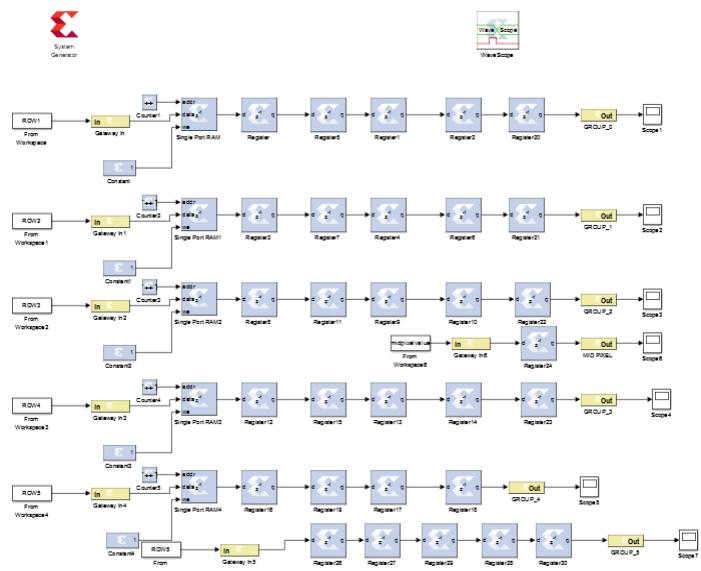


Fig.2 Design concept of register matrix

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 center 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.

B. Photometric component

The sorted pixel groups are provided as the input of the the photometric component. At the output of the photometric filter, pixels are weighted and 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 according to (4). Thus, in parallel to the pixels, the photometric coefficients also have to be processed by the geometric filter in order to obtain the normalization factor defined in (4). 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 . . . 5.

The pixel in the center of the filter window has to be available during the calculation of the required 24 pixel weights. Latching the centered pixel allows the computation of the gray value differences between the centered 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 neighborhood of $-\text{mid_pix}\|$ at one pixel clock signal. All six pipelines are designed identically. At the first internal clock event t_0 , the first pixels of each group are provided to the respective pipeline. At the second internal clock t_1 , the second pixels of each group enter the component. This organization of groups fig.3

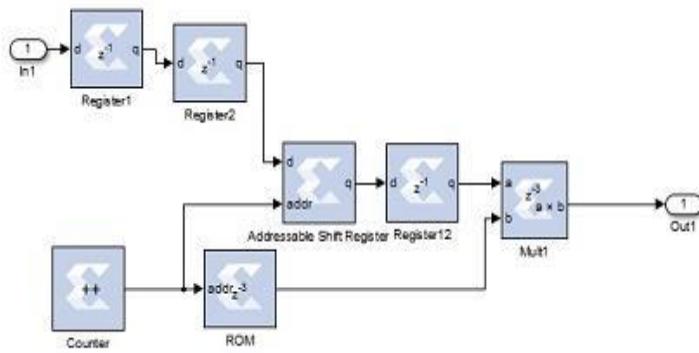


Fig.3 Basic building block of photometric component

allows the processing of the whole filter window in four internal clock cycles corresponding to one pixel cycle

C. Geometric component

For the design of the geometric filter component, advantage is taken of its separability and its symmetry. Because of the separability, 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. Both parts are implemented twice to filter the weighted image data and the photometric weights simultaneously. The input of the vertical component parts is the 2-D array of the filter window and the 2-D array of the corresponding coefficients. Each output is a 1-D vector in which each entry represents one filtered and cumulated column. The coefficients of the geometric component are labeled $-C_0, C_1, C_2$. The output of the geometric filter consists of the filtered unnormalized gray value (kernel result) and the normalization factor (norm result). Due to the symmetry of the weight coefficients of the geometric component, the order of multiplication and addition is swapped in both filter parts. This fact plays an important role in pixel group formation. At first, the weighted gray values which are located at the same distance from the centered pixel

in the filter window are summed up. Because of the equal distance, these gray values should be weighted with the same coefficient anyway. For a 5×5 window, there are always 4 or 8 pixels at the same distance from the centered pixel. 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. 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 coefficients for the geometric component are scaled in such a manner that the sum of the vertical coefficients (and the horizontal ones, respectively) is equivalent to the so-called normalized one. For the signed coefficients with the word length W , the normalized one is equal to 2^{W-1} . This means that the division of the weighted gray

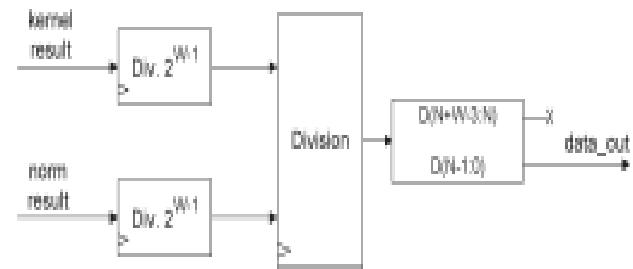


Fig.4 Final normalization of the filtered data

values and photometric coefficients after geometric filtering can be realized as a simple shift operation. In the last stage, the normalized filtered gray value has to be divided by the normalized product of the photometric coefficients. The geometric coefficients are calculated in advance and stored in a block RAM.

D. Normalisation

At the final stage, the kernel result has to be normalized by the norm result as shown in Fig.4. 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 gray 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

IV. IMPLEMENTATION

To accelerate the FPGA design and verification, integrated system-level environment called System Generator from Xilinx is used. Using System Generator, the FPGA design and simulation is carried out using combination of Simulink and Xilinx blocks. The FPGA functional simulation was carried out under MATLAB Simulink environment. After the successful simulation, the synthesizable VHDL code is automatically generated from the models for FPGA synthesis and implementation. During FPGA simulation, the gray scale image data is read sequentially from MATLAB workspace, and the final result also written back into the workspace. After the simulation is completed, a MATLAB program plots the input image and filtered output image for comparison.

ACKNOWLEDGMENT

I would like to thank my guide Mr. Sreejith.S.Nair, assistant professor, department of electronics and communication engineering Jawaharlal college of engineering and technology Palakkad for his unwavering support and valuable suggestions for successful completion of this first phase of the project.

REFERENCES

[1] An FPGA-Based Fully Synchronized Design of a Bilateral Filter for Real-Time Image Denoising Anna Gabiger Rose, Student Member, IEEE, Matthias Kube, Robert

Weigel, *Fellow, IEEE*, and Richard Rose, *Student Member, IEEE*

[2] C. Tomasi and P. Manduchi, —Bilateral filtering for gray and color images,|| in *Proc. IEEE ICCV*, 1998, pp. 839–846.

[3] B. Yan and A.-D. Saleh, —Structure enhancing bilateral filtering of images,|| in *Proc. IEEE PCSPA*, 2010, pp. 614–617

[4] A. Gabiger, M. Kube, and R. Weigel, —A synchronous FPGA design of a bilateral filter for image processing,|| in *Proc. IEEE IECON*, 2009, pp. 1990–1995

[5] L. Costas, P. Colodrón, J. J. Rodríguez-Andina, J. Fariña, and M.-Y. Chow, —Analysis of two FPGA design methodologies applied to an image processing system,|| in *Proc. IEEE ISIE*, 2010, pp. 3040–3044.

[6] N. Sudha and A. R. Mohan, —Hardware-efficient image- based robotic path planning in a dynamic environment and its FPGA implementation,|| *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1907–1920, May 2011.

[7] R. Marin, G. León, R. Wirz, J. Sales, J. M. Claver, P. J. Sanz, and J. Fernández, —Remote programming of network robots within the UJI industrial robotics telelaboratory: FPGA vision and SNRP network protocol,|| *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4806–4816, Dec. 2009 .