IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, March 2017

Real Time Image Denoising using Synchronized Bilateral Filter

Chandni C S[1], Pushpakumari R[2]

PG Scholar, Dept of ECE, Prime College of Engineering, Palakkad, Kerala, India[1]

Assistant Professor, Dept of ECE, Prime College of Engineering, Palakkad, Kerala, India[2]

Abstract: A detailed description of a bilateral filter for image processing is given. The method is non-iterative, local, and simple. It combines gray levels based on both their geometric closeness and their photometric similarity and prefers near values to distant values in both domain and range. The distinctive feature of our design concept consists of processing the entire filter window in one-pixel clock cycle. This feature of the kernel-based design is supported by the arrangement of the input data into groups. Additionally, by the exploitation of the separability and the symmetry of one filter component, the complexity of the design is widely reduced. Combining these features, the bilateral filter is implemented as a highly parallelized pipeline with very economical and effective utilization of dedicated resources. Due to the modularity of the filter design, kernels of different sizes can be implemented with low effort using our design and given instructions for scaling. The resulting image quality depends on the chosen filter parameters only. This filter requires a single image or frame to denoise.

Keywords: Bilateral filter, image denoising, real-time processing, register matrix, photometric filter, geometric filter.

I. INTRODUCTION

Filtering is perhaps the most fundamental operation of image processing and computer vision. In the broadest sense of the term "filtering", the value of the filtered image at a given location is a function of the values of the input image in a small neighborhood of the same location. For example, Gaussian low-pass filtering computes a weighted average of pixel values in the neighborhood, in which the weights decrease with distance from the neighborhood center. Although formal and quantitative explanations of this weight fall-off can be given, the intuition is that images typically vary slowly over space, so near pixels are likely to have similar values, and it is, therefore, appropriate to average them together. The noise values that corrupt these nearby pixels are mutually less correlated than the signal values, so noise is averaged away while thesignal is preserved.Bilateral filtering is a simple, non-iterative scheme for edge-preserving smoothing.

In image processing bilateral filtering has great popularity due to its capability of reducing noise while preserving the structural information of an image. The bilateral filter consists of two components. The nonlinear filter component also called photometric filter preserves detail. Pixels of similar intensity is selected and 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 noise filtering, despite the prevailing view, not always implies resolution reduction but can even be used to sharpen the edges [1] or to enhance the flow like structures [2]. 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 is explained by the author of[3]. The main application of bilateral filtering comes in medical image processing and non-destructive testing.

The major contribution of this paper is the detailed description of thebilateral filter which is implemented in VHDL. The main advantage of this design are the capability of real-time processing and theamount of memory required is less and it is easy to implement.

II. RELATED WORK

A lot of effort has been put into acceleration for use in many practical applicationssince the bilateral filter is used widely. Mainly, two trends can be statedamong the publications concerning speeding up of the bilateral filtering. One is focused on the modification of the filtering components, and another trend is to accelerate the filtering through parallelizing the algorithm or through hardware acceleration.

A fast approximation of the original bilateral filter is proposed in [4]. Here, the 2-D filtering is separated into two 1-D operations for simplicity, performing 1-D bilateral filtering in one arbitrary dimension and filtering the intermediate results. The proportionality of the execution time to the number of filter dimensions decreases from exponential to

IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, March 2017

linear. The results in a filter which is fast enough to be used for pre-processing invideo compression systems but a little memory overhead is required. The image resulting from the modified filter and produced by the original filter have slight differences since the photometric component is not separable.
The approach proposed in [5] has given a basis for numerous extensive works. A numerical scheme for speeding up the filtering via a piecewise linear approximation of the bilateral filter in the intensity domain and substituting the low-pass filtering by down sampling is provided in this acceleration approach. This technique is extended to a 3-D space by transposing the computationpresenting the image intensity as a third dimension of the 2-D image coordinate space is presented in [6]. After that, the authors of [7] formulated the concept of the bilateral grid and implemented the bilateral filter using the proposed data structure on three different graphics processing units (GPUs) and it enables real-timeedge-preserving image manipulation. Not until then, by means of their hardware acceleration, a processing with 30 fps is possible which they assign as real-time performance. Later, the technique in [5] was also implemented on a GPU by the authors of [8] and is also capable of the real-time processing. More recently, the lazy sliding window implementation of the approach in [5] was proposed in [9]. This method is suitable for single instruction multiple data type processors like DSPs which allows performing filtering in the manner efficient both to storage and number of computations.
The idea of using histogram-based approach for accelerating filter instead of a piecewise-linear approximation and subsampling is fast, but a real-time performance of histogram based approach can only be achieved by very large scale integration design of the filter shown in [10]. The authors of [11] report improvement of the speed of segmentation compared with the sequential code based segmentation when implementing an algorithm for color image segmentation for object detection in full parallelism on an FPGA.In [12], Verilog hardware description language code of the design is generated automatically from the models for FPGA synthesis using System Generator from Xilinx.
In [13], a different approach for the FPGA implementation of a real-time bilateral filter has been proposed. This filter is based on the calculation of the filter coefficients from the photometric filter only. Due to the processing of the minimal window of 3×3 and raising of the derived photometric coefficients to the power of 8, elimination of geometric filter is done. The modified bilateral filter can achieve slightly better results compared to the traditional bilateral filter.

III.BASIC THEORY

The bilateral filter is the combination of domain and range filtering. The domain filter averages the nearby pixel values and acts thereby as a low-pass filter. The range filter plays an important part in edge preserving and allows averaging of similar pixel values only, regardless of their position in the filter window. The pixel is skipped, if the value of a pixel in the filter window diverges from the value of the pixel being filtered by a certain amount.
Taking Gaussian noise into account, the shift-variant filtering operation of the bilateral filter is given by

$$\bar{\phi}(\overline{\mathbf{m}}_0) = \frac{1}{k(\mathbf{m}_0)} \sum_{\mathbf{m}\in F} \phi(\mathbf{m}) \cdot s\left(\phi(\mathbf{m}_0), \phi(\mathbf{m})\right) \cdot c(\mathbf{m}_0, \mathbf{m})$$

The term $\mathbf{m}$ = (m, n) denotes the pixel coordinates in the image to be filtered and $\mathbf{m_0}$ = (m_0, n_0) and $\overline{\mathbf{m_0}}$ = (m_0, n_0) represent the coordinates of the centered pixel in the noisy and in the filtered images, respectively. With these notations, φ($\mathbf{m_0}$) means the gray value of the pixel being filtered, and φ($\mathbf{m}$) identifies the gray value of the spatially neighboring pixels to φ($\mathbf{m_0}$) in the filter window F.
The following expressions describe the photometric and the geometric components s(φ($\mathbf{m_0}$), φ($\mathbf{m}$)) and c($\mathbf{m_0}$,$\mathbf{m}$), respectively:

$$s\left(\phi(\mathbf{m}_0), \phi(\mathbf{m})\right) = \exp\left(-\frac{1}{2}\left(\frac{\|\phi(\mathbf{m}_0) - \phi(\mathbf{m})\|}{\sigma_{\mathrm{ph}}}\right)^2\right)$$

$$c(\mathbf{m}_0, \mathbf{m}) = \exp\left(-\frac{1}{2}\left(\frac{\|\mathbf{m}_0 - \mathbf{m}\|}{\sigma_c}\right)^2\right)$$

where parameters σ_{ph} and σ_c regulate the width of the Gaussian curve assigned to s(φ($\mathbf{m_0}$), φ($\mathbf{m}$)) and c($\mathbf{m_0}$,$\mathbf{m}$), respectively.
The photometric component compares the gray value of the centered pixel with the gray values of the spatial neighborhood and computes the corresponding weight coefficients depending on the factor σ_{ph}. The more the absolute difference of the gray values exceeds σ_{ph}, the lower is the corresponding filter coefficient and vice versa. The domain filter c($\mathbf{m_0}$,$\mathbf{m}$) acts as a standard low-pass filter, the weights of which are reciprocally proportional to the spatial distance of the centered pixel to the pixels in the neighborhood.
Normalization guarantees that the range of filtering images does not change significantly due to filtering.

$$k(\mathbf{m_0}) = \sum_{\mathbf{m} \in F} s\left(\phi(\mathbf{m_0}), \phi(\mathbf{m})\right) \cdot c(\mathbf{m_0}, \mathbf{m})$$

Owing to the fact that the coefficients of the photometric component cannot be computed in advance, the division by the normalization factor cannot be avoided by means of prescaling of the filter coefficients.

IV. DESIGN CONCEPT

The bilateral filter architecture consists of three functional blocks. The filter window of 5 x 5 size is chosen for thefiltering operation. The following figure Fig.1 shows the block diagram of functional units.Data_in represents the input data which are read line by line. The register matrix stores and arranges the pixels for further processing. The photometric filter weights the input data according to the gray level value of the processed pixels. The filtering is completed by thegeometric filter and the final output Data_out is the filtered pixel value.
The image data, as well as all constants and coefficients used in the following design concept, are integer numbers.The 5 x 5 window size is the tradeoff between high noise reduction and low blurring effect.

A. Register Matrix

As the intensity of each pixel is different, the intensity dependent photometric filter requires aseparate coefficient calculation for every pixel in the filter window. A filter window of size 5 x 5 is used in this design. Hence for filtering of one image pixel requires the computation of 24 weights.The filter window is moved along the image rows, moving one row down after the previous row has been filtered. Hence five rows have to be stored during the filtering of each row. To avoid the usage of external image buffer, the five input rows are stored in the line storages which are implemented as block RAMs for data with N bits. These five rows include the row to be filtered and the two rows before and after the row to be filtered. This arrangement is shown in Fig 2.
The input data are read into the register matrix in a serial manner. After the center row has been filtered, the “line storage n-2 will move out of the register matrix. At the end of an image row, the remaining four rows are shifted one row down. The line storage n+1 is then processed. The filtered pixels are stored externally.

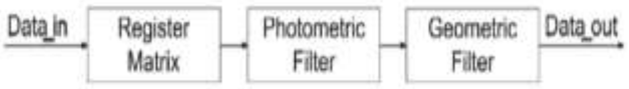

Fig. 1. Functional units of Bilateral filter

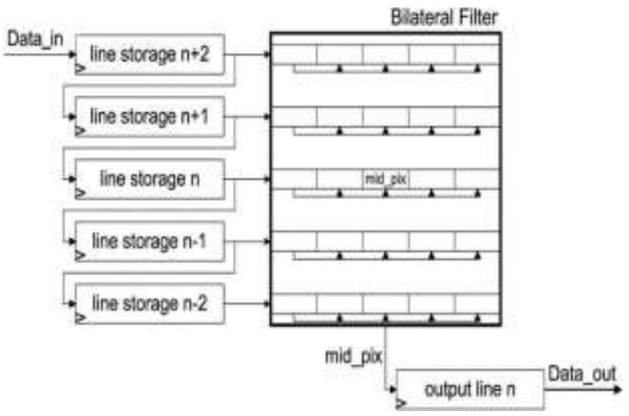

Fig. 2. Principle of the input data retrieval for the image filtering

Processing the pixels in a serial manner delays the operation. The parallel processing of pixels is incorporated in order to speed up the operation and it is achieved by agrouping of pixels in the filter window.

B. Photometric Filter

The grouped pixels are given to the photometric filter. The abstract illustration of the photometric filter component is given in Fig 3. The coefficients, the weighted center pixel, and the weighted pixel values are the outputs. The coefficients are required for the computation of normalization component. This part is used to calculate the difference

between the centered pixel and the neighborhood pixel. Furthermore, there are many operations to be performed in the photometric filter such as division, multiplication, exponential function. To simplify the operation look up table is used. For every possible difference value, the coefficient value is previously calculated and stored in the lookup table. The difference value is the address of the look-uptable and the value stored in the address is the corresponding coefficient value of the difference.
Due to the quantization, the number of the weight coefficientsis limited. This limit depends on three parameters they are the word length N of the input data, parameter σ_{ph} and the word length W of the coefficients. The first point means that increasing the color depth of an image causes a larger amount of intensity differences that have to be stored in the LUT. Depending on the parameter σ_{ph}, the slope of the Gaussian curve is steeper or more flat which influences the number of coefficients different from zero after the quantization. It depends on the word length W itself whose coefficients actually are different from zero after the quantization.

C. Geometric Filter

Geometric filter depends on the spatial distance between the pixels. As the image is two dimensional, 2D filtering is required. Here 2D filtering is replaced by continuous 1D filtering in horizontal and vertical directions. 1D filtering is preferred because of ease of implementation. The overall block diagram is shown in Fig 4. Two horizontal and vertical component parts are used in order to filter the weighted pixel values and the photometric coefficients. The input given to vertical component part is weighted pixel values and the output obtained is five filtered and cumulated column pixel values and is given as input to the horizontal part.
The output obtained at the end of the geometric filter is kernel result and the normalization result. The weighted pixels locates coefficient value at the same distance are added and multiplied by the same. Because of the linearity of the filter, the geometric coefficients can be calculated and stored initially.

D. Normalization

The kernel result and the norm results are the two results obtained from thegeometrical filter. The kernel result is an

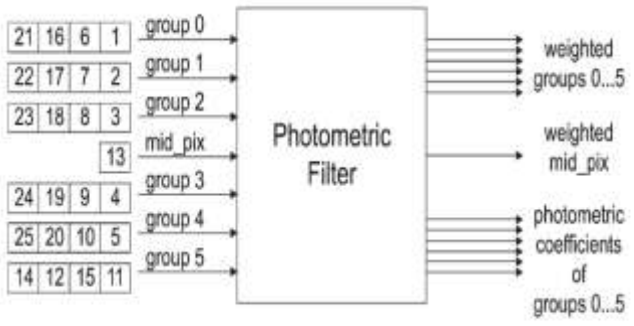

Fig. 3. Input and outputs of photometric filter

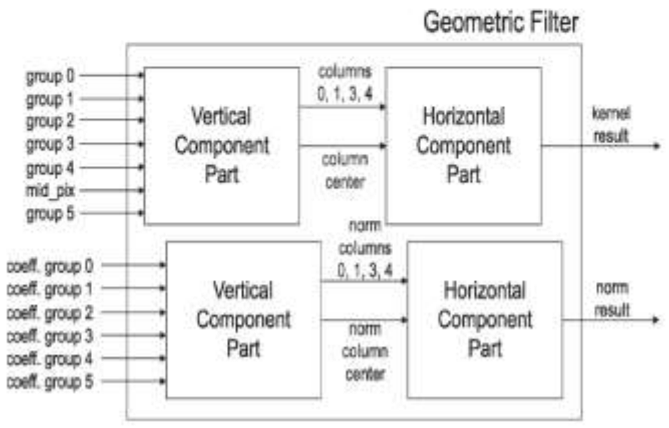

Fig. 4. Abstract illustration of the geometric filter component

unnormalized value and the norm result indicates thenormalization factor. By dividing the unnormalized value bythe normalization factor the normalized value i.e., the filtered pixel is obtained. That is at the final stage, the kernel result has to be normalized by the norm result as shown in fig 5. The final result is forwarded to the external storage.

V. RESULTS

After converting the image to a text file, the bilateral filter is implemented using VHDL and is simulated in ModelSim. The result obtained is a text file which is converted into an image using Matlab.

The resultant image is shown in fig 6 and is a gray scale image with a size of 256 x 256 pixels, so there are 65536 pixel values in total. The original image is shown in left side and filtered image is shown on the right side respectively in fig 6.

VI. CONCLUSION

Simple and easy implementation of bilateral filter for real-time image processing is proposed. The filter design for a kernel size of 5 × 5 shown, which makes it feasible to implement the filter.

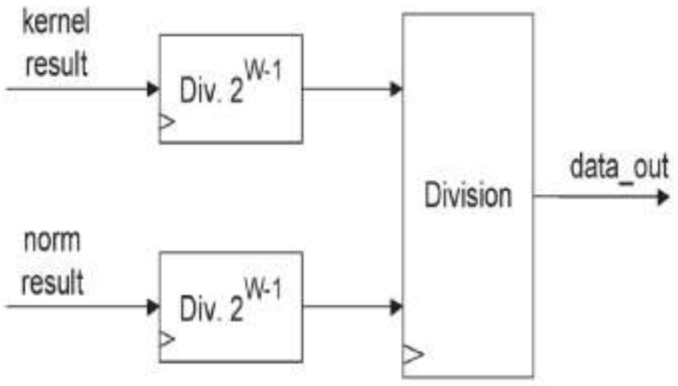

Fig. 5. Final normalization of the filtered data.

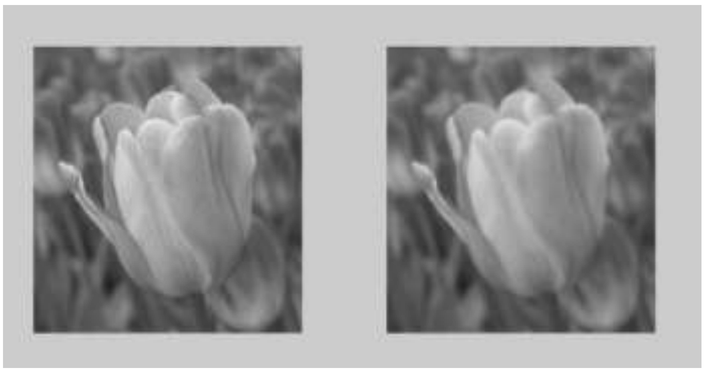

Fig. 6. Resultant image

The introduced register matrix at the first stage of the filter makes external image storage redundant, contributing to thedecrease of the resource demand of the filter implementation. The shown architecture is synchronous and capable of real-time processing supporting high clock frequencies. Conceiving our filter architecture, we kept in mind the scalability of the design in order to enable the implementation of arbitrary filter window size with low effort. The shown filter architecture assures a constant processing delay independent of the filter window size. The total delay is the sum of the processing delay and the fill-up time of the line storages which depends on the kernel size and image width.

ACKNOWLEDGMENT

It is my privilege to express my heartful thanks to my guide **Mrs. Pushpakumari. R**, Assistant Professor, Department of Electronics and Communication Engineering, for her constant encouragement, valuable guidance and timely help for the progress of project work. I also thank all the teaching and non-teaching staffs for the direct and indirect help they rendered.

REFERENCES

[1] B. Zhang and J. P. Allebach, “Adaptive bilateral filter for sharpness enhancement and noise removal,” IEEE Trans. Image Process., vol. 17, no. 5, pp. 664–678, May 2008.
[2] B. Yan and A.-D. Saleh, “Structure enhancing bilateral filtering of images,” in Proc. IEEE PCSPA, 2010, pp. 614–617.
[3] 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.
[4] T. Q. Pham and L. J. van Vliet, “Separable bilateral filtering for fast video preprocessing,” in Proc. IEEE ICME, 2005, pp. 1–4.
[5] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high dynamic- range images,” ACM Trans. Graph., vol. 21, no. 3, pp. 257–266, Jul. 2002.
[6] S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach,” in Proc. ECCV, 2006, pp. 568–580.
[7] J. Chen, S. Paris, and F. Durand, “Real-time edge-aware image processing with the bilateral grid,” ACM Trans. Graph., vol. 26, no. 3, pp. 1–9, Jul. 2007.
[8] Q. Yang, K.-H. Tan, and N. Ahuja, “Real-time O(1) bilateral filtering,” in Proc. IEEE CVPR, 2009, pp. 557–564.
[9] M. M. Bronstein, “Lazy sliding window implementation of the bilateral filter on parallel architectures,” IEEE Trans. Image Process., vol. 20, no. 6, pp. 1751–1756, Jun. 2011.
[10] Y.-C. Tseng, P.-H. Hsu, and T.-S. Chang, “A 124 Mpixels/sec VLSI design for histogram-based joint bilateral filtering,” in IEEE Trans. Image Process., Nov. 2011, vol. 20, no. 11, pp. 3231–3241.
[11] H. Zhuang, K.-S. Low, and W.-Y. Yau, “Multichannel pulse-coupled neural-network-based color image segmentation for object detection,”IEEE Trans. Ind. Electron., vol. 59, no. 8, pp. 3299–3308, Aug. 2012.
[12] C. Charoensak and F. Sattar, “FPGA design of a real-time implementation of dynamic range compression for improving television picture,” in Proc. IEEE ICICS, 2007, pp. 1–5.
[13] T. Q. Vinh, J. H. Park, Y.-C. Kim, and S. H. Hong, “FPGA implementation of real-time edge-preserving filter for video noise reduction,” in Proc. IEEE ICCEE, 2008, pp. 611–614.
[14] M. Zhang and B. K. Gunturk, “Multiresolution bilateral filter for image denoising,” IEEE Trans. Image Process., vol. 17, no. 12, pp. 2324–2333, Dec. 2008.
[15] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.