# Denoising of an Image using Bilateral Filter on FPGA

C . Pallavi
M.Tech –ECE(VLSI)
Vaagdevi Institute of Technology & Science(VITS)
JNTUA
Proddatur , India

S . Siddeswara Reddy
Assistant Professor in ECE
Vaagdevi Institute of Technology & Science
JNTUA
Proddatur , India

***Abstract:*** **A bilateral filter for image processing is implemented on synchronous field programmable gate array. The reason for selecting bilateral filter is that it reduces noise while maintaining the details of the image and the design of bilateral filter is described on register transfer level. Our design concept consists of changing the clock domain in such a way that only one pixel clock cycle is required for the processing of the entire filter window i.e, kernel-based processing is performed. The main purpose of this kernel-based design is that it divides the input data into groups so that the internal clock of the design is a multiple of the pixel clock. Finally, the difficulty of the design is mostly reduced by making use of the separability and regularity of one filter component. By utilizing the dedicated resources effectively and economically the bilateral filter is implemented as highly parallelized pipeline structure. With low effort different sizes of kernels can be implemented because of the modularity of the filter design. Negligible quality loss of the output image is obtained due to the quantization of filter coefficients.**

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

## I.INTRODUCTION

A filter that has the ability of reducing noise while maintaining the structural information of an image is bilateral filter. Due to this feature the bilateral filter has become more popular and is mostly used in image processing. Bilateral filtering is a technique to smooth images while preserving edges. The bilateral filter [1] consists of two components. They are: Photometric filter and Geometric filter. 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 noise is reduced by the selective averaging and the amount of the blurring is reduced by the low-pass filtering.

Adaptive bilateral filter for sharpness enhancement and noise removal[2] can be used to sharpen the edges and also performs noise filtering. Structure enhancing bilateral filtering of images[3] is used to enhance the flow like structures. Noise reduction and adaptive contrast enhancement for local tone mapping[4] applies bilateral filter for noise reduction in a method for local tone mapping which maps high dynamic range image to low dynamic range image. A synchronous FPGA design of a bilateral filter for image processing[5] achieves very satisfying results considering detail preservation. Design methodologies based on hardware description languages[6] provides the possibility of direct specification of the clocking scheme.

Separable bilateral filtering for fast vedio preprocessing [7] presents a fast approximation of the original bilateral filter where the 2-D filtering is separated into two 1-D operations performing 1-D bilateral filtering in one arbitrary dimension and filters the intermediate result in the same manner in the subsequent dimension. It decreases the execution time to the number of filter dimensions from exponential to linear. This approach requires a little memory overhead but results in a filter which is fast enough to be used for pre-processing in video compression systems. However, as the photometric component of the bilateral filter is not separable, the image resulting from the modified filter is documented to be slightly different from the image produced by the original filter. A 124 Mpixels/sec VLSI design for histogram-based joint bilateral filtering [8] shows that the histogram-based approach can only be achieved by very-large-scale integration design of the filter. The memory demand of the histogram-based acceleration method is also high but is lower than that of the piecewise-linear approximation and subsampling approach. We choose an FPGA implementation for their image processing system because moving time critical functionalities, like the edge detection in an image, to hardware platforms makes it possible to keep delays in the control loop to a minimum[9]. [10] and [11] presents excellent experience of using FPGAs for motion control of robots based on real-time image processing. The main reason for using FPGAs for real-time robotics tasks is the ability of FPGAs to satisfy the requirement for high computational power and data throughput [11]. Moreover, FPGA solutions offer additional advantages, such as reconfigurability and portability. However, in fields of applications in which high power efficiency is crucial, an FPGA solution is preferable. Features, design tools and application domains of FPGAs[12] emphasizes that FPGA based digital processing systems achieve better performance, at a lower cost, than traditional solutions based on DSPs and the advantage of using FPGA even at both high complexity and timing constraints.

The major parallel to our design consists in implementing the bilateral filter on an FPGA without any modification. This approach is sometimes called brute-force method. However, the main difference to our work is that it is developed using an high level synthesis(HLS) tool. The resulting architecture presents a 3 × 3 filter kernel. In contrast, our design is based on an RTL description and presents a 5 × 5 filter kernel. Our design allows high clock frequency and high data throughput and shows only a slight increase of resource demand considering the larger kernel. From this we can say that our architecture utilizes hardware resources more efficiently and more economically.

## II. BILATERAL FILTER

The basic idea underlying bilateral filtering is to do in the range of an image what traditional filters do in its domain. Two pixels can be close to one another, that is, occupy nearby spatial location, or they can be similar to one another, that is, have nearby values, possibly in a perceptually meaningful fashion. The bilateral filter [1] represents the idea of combining two filter components, the domain and range filter components. First stage is register matrix followed by photometric filter and geometric filter.

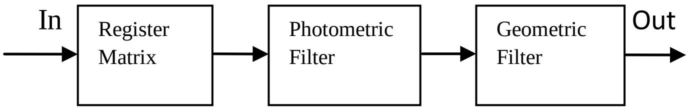

Fig.1. Functional units of bilateral filter

In our design concept, the bilateral filter utilizes the resources effectively and is realized as highly parallelized pipeline structure. The selected window size is 5 × 5 and is the tradeoff between high noise reduction and low blurring effect. The bilateral filter is divided into three blocks to reduce the difficulty. In Fig.1. "In" is used for reading the input data line by line and the input data is given to register matrix. The second stage of bilateral filter is photometric filter. The purpose of this filter is that the input data is weighted according to the intensity of the processed pixels. The third stage in the bilateral filter is the geometric filter. The photometric coefficients are processed by the geometric filter and it has the property of separability and symmetry of filter component that reduces the complexity of the design. "Out" is used to represent filtered data of the image processed by the three stages of the bilateral filter.

The photometric filter is also referred as range filter and the geometric filter is also referred as domain filter. The domain filter averages the nearby pixel values and acts thereby as a low-pass filter. The range filter stands for the nonlinear component and plays an important part in edge preserving. This component allows averaging of similar pixel values only, regardless of their position in the filter window. If the value of a pixel in the filter window diverges from the value of the pixel being filtered by a certain amount, the pixel is skipped.

A bilateral filter is a non-linear, edge-preserving and reduces noise and also smoothes images. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g. range differences, such as color intensity, depth distance, etc.). This preserves sharp edges by systematically looping through each pixel and adjusting weights to the adjacent pixels accordingly.

Taking Gaussian noise into account, the shift-variant filtering operation of the bilateral filter is given by

$$\emptyset^{\bar{}(m\bar{}0)} = \frac{1}{k(m0)} \sum_{m \in F} \emptyset(m).\, s\big(\emptyset(m0), \emptyset(m)\big).\, c(m0, m) \qquad (1)$$

The term m = (*m*, *n*) denotes the pixel coordinates in the image to be filtered and m0 = (*m*0, *n*0) and m̄0 = ($\bar{m}$0, $\bar{n}$0) represent the coordinates of the centered pixel in the noisy and in the filtered images, respectively. With these notations, ∅̄(m̄0) means the gray value of the pixel being filtered, and ∅(m) identifies the gray value of the spatially neighboring pixels to ∅ (m0) in the filter window *F*. The following expressions (2) and (3) describe the photometric and the geometric components *s*( ∅ (m0), ∅ (m)) and *c*(m0,m), respectively:

$$\mathrm{s}\big(\emptyset(m0), \emptyset(m)\big) = exp\left(-\frac{1}{2}\left(\frac{\|\emptyset(m0) - \emptyset\ (m)\|}{\sigma p\, h}\right)^{\wedge}2\right) \qquad (2)$$

$$\mathrm{c}(m0, m) = exp\left(-\frac{1}{2}\left(\frac{\|m0 - m\|}{\sigma c}\right)^{\wedge}2\right) \qquad (3)$$

where parameters *σ* ph and *σ* c regulate the width of the Gaussian curve assigned to *s*(∅(**m**0), ∅(**m**)) and *c*(**m**0,**m**), respectively. The photometric component compares the gray value of the centered pixel with the gray values of the spatial neighbourhood 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*(**m**0,**m**) acts as a standard low-pass filter, the weights of which are reciprocally proportional to the spatial distance of the centred pixel to the pixels in the neighborhood.

Normalization with

$$k(m0) = \sum_{m \in F} s\big(\emptyset(m0), \emptyset(m)\big).\, c(m0, m) \qquad (4)$$

guarantees that the range of the filtered images does not change significantly due to the filtering.

As range parameter increases, the bilateral filter gradually approaches Gaussian convolution more closely because the range Gaussian widens and flattens, which means that it becomes nearly constant over the intensity interval of the image. On increasing the spatial parameter, the larger features get smoothened. Two parameters of interest in Bilateral filter are σs(geometric) and σr(range) . These parameters are not optimal for all images. In this approach, the values of σs and σr are chosen as 1 and 60respectively.

A. *The Qualities of Bilateral Filter are*

- Its formulation is simple: each pixel is replaced by a weighted average of its neighbors. This aspect is important because it makes it easy to acquire intuition about its behavior, to adapt it to application-specific requirements, and to implement it.

• It depends only on two parameters that indicate the size and contrast of the features to preserve.

• It can be used in a non-iterative manner. This makes the parameters easy to set since their effect is not cumulative over several iterations.

• It can be computed at interactive speed even on large images, and even in real time if graphics hardware is available.

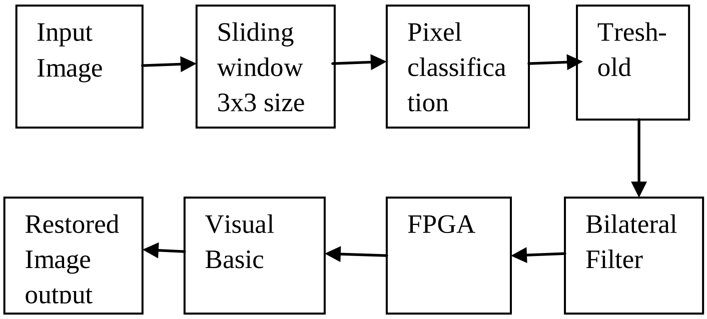

Fig.2. Block diagram

The above block diagram mainly consists of taking at 256/256 image and adding noise to it and converting a noise image into a matrix and then dividing a entire matrix into 3×3 siding window, after dividing that window the sliding window is applied with our Bilateral filter to eliminate the noise pixel. This is represented as 0 and 255 after the eliminating noise pixel by images convert into text format. The text format is converted into bit stream. which is compatible to a FPGA kit is used by Spartan 3E which consist of micro blaze processor which is 32 bit processor consisting of RISC(Reduced Instruction Set) architecture. The output of FPGA kit is visualized in visual basic where the noise less image is retrieved.

*B. Applications of Bilateral Filter*

*1. Denoising* - This is the original, primary goal of the bilateral filter, where it found broad applications that include medical imaging, tracking, movie restoration.

*2. Contrast Management* - Bilateral filtering has been particularly successful as a tool for contrast management tasks such as detail enhancement or reduction. The bilateral filter is used to separate an image into a large-scale component and a small-scale component by subtracting filtered results. Texture and Illumination Separation, Tone Mapping, Retinex, and Tone Management Bilateral filtering an image at several different settings decomposes that image into large-scale/small-scale textures and features. These applications edit each component separately to adjust the tonal distribution, achieve photographic stylization, or match the adjusted image to the capacities of a display device.

*3. Depth Reconstruction* - The bilateral filter is applied in stereo reconstruction, the recovery of depth values from correspondences between pixels different views. Ideally we wish to find a corresponding point in the right image for every pixel in the left image. As the distance between these point pairs, the disparity, is inversely proportional to the depth at that pixel, this information is equivalent to recovering the scene geometry. To pair the pixels with points in the other image, stereo algorithms typically compute a similarity score such as color differences or local correlation. Locally aggregating these scores using bilateral weights significantly improves the accuracy and reduces noise in the recovered depth maps.

*4. Data Fusion* - These applications use bilateral filtering to decompose several source images into components and then recombine them as a single output image that inherits selected visual properties from each of the source images.

*5. 3D Fairing* - In this counterpart to image denoising, bilateral filtering applied to 3D meshes and point clouds smooths away noise in large areas and yet keeps all corners, seams, and edges sharp.

*6.Video Stylization* – We can iterate the bilateral filter to simplify video content and achieve a cartoon look. They demonstrate that the bilateral filter can be computed in real time at video resolution using the numerical scheme on modern graphics hardware. Later, the bilateral filter is ported on the GPU using the bilateral grid and achieved similar results on high-definition videos. The bilateral filtering is an effective preprocessing for edge detection: filtered images trigger fewer spurious edges. To modulate the smoothing strength of the bilateral filter, they modify it to control the degree of edge preservation.

*C.Algorithm*:

Step 1: Select 2-D window of size 3**X** 3. Assume that the pixel being processed is f(x,y).

Step 2: If $0 < f(x,y) < 255$ ,then is an uncorrupted pixel and its value is left unchanged.

Step 3:All the pixels in the selected window are 0sor255s or both then two cases occur.

Case i): If the selected window contain all the elements as 0's and 255's. Then replace with the mean of the element of window and obtain the denoised image.

Case ii): Check each pixel in the selected window is noise free, if noise free then find range of the image, else set range=0

Step4:Apply bilateral filter.

Step5: Denoised image is obtained.

## III. EVALUATION OF QUALITY OF IMAGE

The accuracy and the performance of the image quality are calculated by using PSNRdb and MSSIM.

1)PSNRdB: Peak signal-to-noise ratio, often abbreviated PSNR. It is computed as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. PSNR is most easily defined via the mean squared error (*MSE*). Given a noise-free $M \times N$ monochrome image $I$ and its noisy approximation $K$, *MSE* is defined as:

$$MSE = \frac{1}{MN}\sum_M \sum_N [Q(i,j) - Q^{-}(i,j)]^2 \qquad (5)$$

peak-signal-to-noise ratio is measured in decibels and is given as:

$$\text{PSNRdB} = 20.\log10\left(\frac{255}{\sqrt{MSE}}\right) \qquad (6)$$

where MSE represents the mean squared error between the image to be compared and the reference image. M represents the width of the image, N represents the height of the image and i,j are pixel positioning coordinates. The noiseless M × N image with gray values Q (i,j) provides the reference for the measurement of the MSE. The gray values $Q^{-}$(i,j) originate from the image to be compared. 255 represents the maximum gray value depending on the word length after the digitalization of the images. Considering the quality of the noise, PSNRdB has the ability to suppress noise regardless of the perceived visual quality of the filtered image. Even though PSNR is expressed in decibels, it is dimensionless as both the numerator and the denominator are pixel values. PSNR is popular for several reasons. It is fast to compute, and simple to understand.

2) MSSIM: The **structural similarity** (SSIM) index is a method for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio(PSNR) and mean squared error(MSE), which have proven to be inconsistent with human eye perception. The difference with respect to other techniques mentioned previously such as MSE or PSNR is that these approaches estimate perceived errors; on the other hand, SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. The mean structural similarity index MSSIM is a method for the evaluation of the image quality that takes advantage of the characteristics of the human visual system. First, the local structural similarity SSIM of the 11 × 11 image blocks **v**(∅ref ) and **v**($∅^{-}$) is calculated as

$$SSIM\big(V(\emptyset ref), V(\emptyset^{-})\big) = l(V(\emptyset ref), V(\emptyset^{-})).\ c\big(V(\emptyset ref), V(\emptyset^{-})\big).s\big(V(\emptyset ref), V(\emptyset^{-})\big) \qquad (7)$$

where $l$(**V**( ∅ ref ),**V**( $∅^{-}$ ) is the luminance comparison function, $c$(**V**(∅ref),**V**($∅^{-}$ )) compares the contrast of the image blocks after luminance subtraction, & $s$(**V**( ∅ ref ), **V**( $∅^{-}$) conducts the structure comparison after contrast normalization. After averaging the SSIM of $J$ blocks over the whole image, the mean value MSSIM is of an entire image represented by $∅^{-}$ $i$s identified.

$$MSSIM(\emptyset ref, \emptyset^{-}) = \frac{1}{J}\sum_{j=1}^{J} SSIM\big(Vj(\emptyset ref), V(\emptyset^{-})\big) \qquad (8)$$

If MSSIM = 1 means that two images are completely identical. If the MSSIM value is smaller , then less structural similarity is shown between two images.

## IV. RESULTS

The bilateral filter was simulated in matlab and then it was implemented on FPGA. The filtered images was compared by psnr values to estimate the image quality drop due to quantization of the filter coefficients.

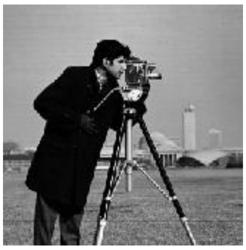

Fig.3. Input Image

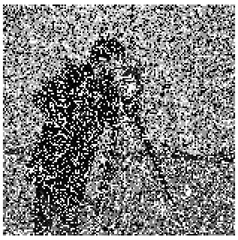

Fig.4. Noisy Image

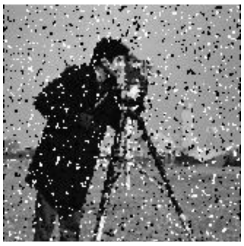

Fig.5. Median Filtered Image

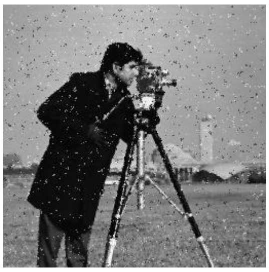

Fig.6.Adaptive Median Filtered image

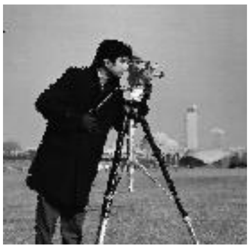

Fig.7.DBA Filtered image

First the given image is cameraman and to this image a noise density of 0.5 is added. To the noise image we applied median filter. By applying median filtering the noise is reduced to some extent but still there is some amount of noise present in the image. So, we applied DBA algorithm to reduce noise. DBA is abbreviated as decision based algorithm. By using DBA algorithm the noise present in the image is reduced.

The filtered images are compared by using psnr values. The noise density added to given image is 0.5. This noisy image has apsnr value of 8.09dB. The median filtered image has the psnr value of 14.34dB and the DBA filtered image has the psnr value of 24.45dB. The Fig.7. shows that the noise is reduced by using DBA algorithm. But still there is some amount of noise present in the image and it is filtered by using bilateral filter as shown in Fig.8.

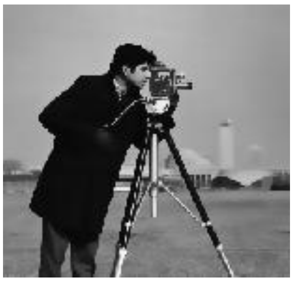

Fig.8. Bilateral Filtered Image

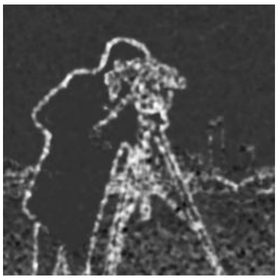

Fig.9 MSSIM of bilateral filter

To input image we applied bilateral filter. The bilateral filtered image has the psnr value of 32.64dB. Fig.6. shows that the noise is reduced by using bilateral filter. For filtering in Matlab, no quantization of the filter coefficients was applied. The filter is implemented in matlab and the psnr values are calculated to estimate the quality of the image. The Fig.9. shows the ssim of the bilateral filtered image.

TABLE 1

Comparison of PSNR Values of Different Algorithms For Cameraman Image at Different Noise Densities

| Noise In % | Noised image | MF | AMF | DBA | BF |
|---|---|---|---|---|---|
| 10 | 15.12 | 25.98 | 28.83 | 33.96 | 34.64 |
| 20 | 12.09 | 23.69 | 26.58 | 30.58 | 31.15 |
| 30 | 10.32 | 20.82 | 24.69 | 28.12 | 29.52 |
| 40 | 9.08 | 17.49 | 21.33 | 26.25 | 28.71 |
| 50 | 8.08 | 14.28 | 19.15 | 24.10 | 27.65 |

The above table1 shows that the bilateral filter has highest psnr value when compared with median filter(MF), adaptive median filter(AMF) and decision based algorithm(DBA).

*Implementation on FPGA:*

A Virtex-5 FPGA platform equipped with a SPATAN 3EDK device was used. The FPGA board was connected to a camera with a 12-b resolution depth, generating 30 fps at a full resolution of 1024 × 1024 pixels. 30fps is the maximally achievable frame rate.

The maximal data flow reaches approximately 31.5 Mpixel/s. Consequently, we restricted the clock frequency of our design to 40 MHz in this application. The internal clock frequency is 160 MHz. With this clock rate, a maximal throughput of 38 fps is possible. The maximum possible internal frequency is 220MHz. Maximal operating frequency is 55MHz.

Considering the image resolution of 1024 × 1024 pixels, the following frame rate can be computed:

$$\left[(1024 \times 1024)\frac{pixels}{frame}\cdot\frac{18.18ns}{pixel}\right]\text{^-1=52.45}\frac{frames}{second}$$

This calculation is valid only for a throughput of 1 pixel/cycle by using our design.

The total delay of the output pixels of our architecture with a kernel size of 5 × 5 pixels applied to an image of 512 × 512 pixels is 2560 + 36 cycles. The time required for filling up of the register matrix, depending on the kernel size and image width, results in a delay of 5 × 512 = 2560 cycles. The processing time still accounts for 36 cycles. The normalization by division costs 24 cycles, which makes out 66% of the whole processing time.

Our design works at the highest clock frequency. Considering the kernel size of 5 × 5 pixels and the switching of the time domain, our architecture presents only the third highest frame rate. However, it looks different if we implement a 3 × 3 filter kernel. In this case, the operating frequency is 110 MHz, and the resulting frame rate doubles. An efficient parallel implementation of a bilateral filter for a 5×5 mask requires 25 multipliers. Our design efficiently requires only 23 multipliers that are obtained in the synthesis report. Therefore, considering the implemented window size of 5 × 5 pixels, we use the resources more economically.

## V. ADVANTAGES OF BILATERAL FILTER

- Easy to understand – Weighted mean of nearby pixels
- Easy to adapt – Distance between pixel values
- Easy to set up – Non-iterative

## VI. CONCLUSION

The bilateral filter with a kernel size of 5 × 5 for real-time image processing is implemented on medium-sized FPGA that supports high clock frequencies and utilizes the resources effectively and economically. The first stage of filter is register matrix and it decreases the resource demand of the filter implementation. Arbitrary filter window size is implemented with low effort and with constant processing delay. 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. At high noise density, PSNR value of the proposed algorithm is almost 2 to 4dB better than the existing algorithms and evaluated in terms of PSNRdB.

## REFERENCES

1. C. Tomasi and P. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE ICCV*, 1998, pp. 839–846.
2. 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.
3. B. Yan and A.-D. Saleh, “Structure enhancing bilateral filtering of images,” in *Proc. IEEE PCSPA*, 2010, pp. 614–617.
4. 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.
5. 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.
6. T. Riesgo, Y. Torroja, and E. de la Torre, “Design methodologies based on hardware description languages,” *IEEE Trans. Ind. Electron.*, vol. 46, no. 1, pp. 3–12, Feb. 1999.
7. T. Q. Pham and L. J. van Vliet, “Separable bilateral filtering for fast video preprocessing,” in *Proc. IEEE ICME*, 2005, pp. 1–4.
8. 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.
9. 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.
10. 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.
11. 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.
12. J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, “Features, design tools, and application domains of FPGAs,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1810–1823, Aug. 2007.
13. H. Dutta, F. Hannig, J. Teich, B. Heigl, and H. Hornegger, “A design methodology for hardware acceleration of adaptive filter algorithms in image processing,” in *Proc. IEEE ASAP*, 2006, pp. 331–340.
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.
16. A. Gabiger-Rose, M. Kube, P. Schmitt, R. Weigel, and R. Rose, “Image denoising using bilateral filter with noise-adaptive parameter tuning,” in *Proc. IEEE IECON*, 2011, pp. 4515–4520.