VLSI Implementation of Modified Guided Filter for Edge Preservation
Ms.M.Muthuselvi¹ Mr.V.Suresh²
¹P.G Student ²Assistant Professor
1,2Department of Electronics & Communication Engineering
1,2Maha Barathi Engineering College- Chinnasalem, Tamilnadu, India

Abstract—Filtering is widely used in the image and video processing application for various technology. Field Programmable Gate Array (FPGA) technology has become a viable target for the implementation of real-time algorithms is suitable for the video image processing applications. In this paper a explicit image filter called guided filter is proposed to remove the noise in images smoothing and sharpening of images. We proposed the systolic array architecture for modified guided filter. The guided filter is derived from the local linear model computing the filtered output by considering the content of guided image can be the input image or different image. The guided filter can be used as the edge preserving smoothing operator. Transfer the structure of filtering image to the filtering output enabling and filtering application like dehazing and guided feathering. Computer vision and computer graphics applications, include edge-aware smoothing, HDR compression, image matting. Avoid halo effect using the weighted least square filter. FPGA can be used to implement any logical function that an ASIC could perform. VLSI Implementation for modified guided filter to achieve the high speed, low power consumption and reduce area.

Key words: Edge preserving filtering, linear time filtering, weighted least square filter

I. INTRODUCTION

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing.

FPGA based hardware accelerators have become more and more important for bioinformatics applications. These applications use wide range of algorithms, including searches in large databases, sequence alignment, statistical analysis and image processing. A part of these algorithms can be efficiently accelerated using FPGA devices.

Biological and biomedical experiments like microarray experiments, high-content screening (HCS) or cellular microscopy result in large amount of image data. These data have to be processed to evaluate the experiments. There are different applications available for this purpose, for example the Cell Profiler, which can process and analyze cellular microscopy images. The final goal would be an FPGA based hardware accelerator for (biomedical) image processing purposes. This would include several reconfigurable execution units, which would enable the parallel processing of the data.

A. Filtering:
Transform pixel intensity values to reveal certain image characteristics.
1) Enhancement: Improves contrast
2) Smoothing: Removes noise
3) Template matching: Detects known patterns

B. Image Filtering:
Images are often corrupted by random variations in intensity, illumination, or have poor contrast and can’t be used directly.

C. Noise:
Images are corrupted by random variation in intensity values called noise due to non-perfect camera acquisition or environmental conditions.

D. Types of Noises:
1) Salt and pepper noise: Random occurrences of both white and black intensity values.
2) Impulse noise: Random occurrences white intensity values.
3) Gaussian noise
4) Shot noise
5) Quantization noise
6) Film grain

This project introduces a reconfigurable test system and two basic image processing functions implemented with FPGA. These functions are a 3 x 3 median filter and a 3 x 3 sobel filter, which can process 8 bit grayscale images. Advanced RISC microprocessors can solve complex computing tasks through a programming paradigm, based on fixed hardware resources. For most computing tasks it is cheaper and faster to develop a program in general-purpose processors (GPPs) specifically to solve them. These costs involve three parts:

1) Hardware Costs:
GPPs are larger and more complex than necessary for the execution of a specific task. Developing application-specific processors for highly specialized algorithms is warranted only for large-volume applications that may require high power efficiency at expense of great hardware design cost;

2) Design Costs:
Functional units that may be rarely used in a given application may be present in GPPs, and may consume substantial part of the design effort;

3) Energy Costs:
Too much power is spent with functional units or blocks not used during a large fraction of the processing time. For specific applications or demanding requirements in terms of power, speed or costs, one may rely on either dedicated
in a pure hardware model, a given algorithm is converted into a single hardware description, which is loaded into the FPGA. There is no relevant contribution of this model to reconfigurable architectures, since the configuration is fixed at design time.

4) **Statically Reconfigurable Design (SRD):**
The circuit has several configurations (N) and the reconfiguration occurs only at the end of each processing task. This can be classified as runtime reconfiguration, depending on the granularity of the tasks performed between two successive reconfigurations. In this way, the programmable devices are better used and the circuit can be partitioned, aiming for resources reusability.

5) **Dynamically Reconfigurable Design (DRD):**
The circuit also has N configurations, but the reconfiguration takes place at runtime. This kind of design uses more efficiently the reconfigurable architectures. The timing overhead associated to this RTR procedure has to be well characterized within the domain of the possible set of runtime configurations.

The overall performance wills be determined by the overhead-to-computing ratio. The implementation may use partially programmable devices or a set of conventional programmable devices. The resultant architecture is called DRA. The image data, as well as all constants and coefficients used in the following design concept, are integer numbers. As discussed in Section VI, there is no need to implement floating-point computation. With the aid of the presented design concept, the Guided filter can be realized as a highly parallelized pipeline structure giving great importance to the effective re-source utilization.

![Fig. 1: Order of the functional units of the Guided filter](image1)

The design concept for the implementation of the Guided filter is subdivided into three functional blocks. The block-based design approach reduces design complexity and simplifies validation. Fig.1.1 presents these units and their order in the concept. The input data marked by “Data_in” are read line by line and arranged for further processing in the register matrix. The second unit is the photometric filter which weights the input data according to the intensity of the processed pixels. The filtering is completed by the geometric filter, and the filtered data are marked by “Data_out.”

In the following: section II related work. Design of hardware architecture in section III. While Section IV presents the proposed hardware implementation. Next, Section V shows results and comparison with related work. Finally, Section VI concludes this paper.

**II. RELATED WORK**

In recent years, a fair amount of work has been carried out on real-time hardware implementations of local stereo matching algorithms [13]; a thorough review is presented in [3]. The majority of these implementations have adopted simple fixed support and multiple window methods, therefore trading accuracy for speed. High matching accuracy though is of foremost importance in many of today’s embedded vision applications. As such, a few attempts have been made recently directed towards improving the matching accuracy, either by combining different stereo algorithms together, or by implementing modified versions of SGM and ADSW algorithms. The hardware implementation in performs a modified version of the Census transform in both the intensity and gradient images, in combination with the SAD correlation metric. An FPGA implementation of a stereo algorithm based on the neural network and Disparity Space Image (DSI) data structure is introduced. The real-time FPGA-based stereo matching design presented in combines the mini-Census transform and the Cross-based cost aggregation. SGM-based stereo matching systems have been introduced in [5], [6] and implemented on FPGAs and a hybrid FPGA/RISC architecture, respectively. The technical details/parameters of the different implementations are summarized.

The works that are closely related to ours in terms of the matching algorithm are the works in [7], [8], [9], which implement ADSW-based algorithms. [7] proposed the VLSI design of a hardware-friendly ADSW algorithm that adopted the mini-Census transform to improve the accuracy and robustness to radiometric distortion. [8] proposed the implementation of a complete stereo vision system, which incorporates an ADSW algorithm and also integrates pre- and post-processing units. Finally, a hardware-oriented stereo matching system based on the adaptive Census transform is presented in [9]. The aforementioned high-quality ADSW-based systems follow a similar algorithm-to-hardware mapping methodology. That is, a complex, but accurate, algorithm is adapted for dedicated hardware implementation through a series of algorithmic modifications/approximations. In most cases, however, these implementations scarily part of the accuracy; quality reduction compared to the original implementation of the ADSW approach in [4] is ~ 4-5%.

In contrast, the proposed stereo matching architecture implements the ADSW aggregation step in a different way; by smoothing the SCV with an edge-preserving filter, the GIF. Due to this type of filter, and its optimized hardware design presented in this work, the proposed architecture pushes further the accuracy limits of hardware-based stereo matching systems, while it also achieves real-time frame-rates for HD images.

**III. DESIGN OF HARDWARE ARCHITECTURE**

![Fig. 2: Folding Architecture](image2)
A. Buffer:
A buffer is a data area shared by hardware devices or program processes that operate at different speed or with different sets. The buffer allows each device or process to operate without being held. The purpose of a buffer is to hold data right before it is used.

B. Address Decoder:
Address decoder is a decoder circuit that has two or more bits of an address bus as inputs and that has one or more device selection lines as output.

An address decoder with n address input bits can serve up to 2^n separate devices. Several members of the 7400 series of integrated circuit are address decoder.

C. Multiplexer:
A multiplexer of 2^n inputs has n select line, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth.

D. Finite State Machine:
A state machines is any device that store the status of processing at a given time and can operate on input to change the status and /or cause an action or output to take place for any given change.

E. Read Only Memory:
RAM is a type of “build-in” memory that is used with computers and other electronics devices. ROM may only be read, it is either modified with extreme difficulty or not at all.

IV. PROPOSED MODIFIED GUIDED FILTER

In this section, we first describe how the GIF, the core element of the proposed stereo matching architecture, is implemented in hardware efficiently in a way that its logic resources are independent of the kernel radius r. Guided filter can have an efficient dedicated hardware design, as the basic operation involved is the mean filter with windows of radius r. The mean intensity of pixels over rectangular windows in the image can be implemented in a fast way using the integral image technique. However, this technique requires huge amount of memory, especially for high-resolution images. Therefore, we instead followed a variant of the approach in to implement a custom mean filter design that consumes compact hardware resources. The main idea is to maintain a sum for each column in the image to be filtered. Each column sum accumulates 2r+1 pixels, while the window sum is computed by adding 2r+1 adjacent column sums. While filtering the image, the window sum is updated using the two-step approach. When the window is moved to the right from one pixel to the next, the column sum to the right of the window is yet to be computed for the current row, so it is centered one row above. Therefore, the first step consists of updating the column sum to the right of the window, by subtracting its topmost old pixel and adding one new pixel below it. The second step moves the window to the right and updates the window sum by subtracting its leftmost column sum (old column sum), and adding the updated column sum computed in step 1 (new column sum). Systolic array architecture is used to increase the speed of guided filtered. The 9 stage pipeline architecture is proposed in this paper.

Fig. 3: Block Diagram of Modified Guided Filter

V. BLOCK DIAGRAM DESCRIPTION

A. Input Image:
Grayscale digital image is an input image in which the each value of pixel is a single sample that is carries only intensity information.

B. Register Matrix:
The photometric filter component, also often referred to as a range filter in the related literature, is a nonlinear filter. It means that the filter coefficients change for every filter position. Thus, the pixel weights for the photometric component have to be calculated separately for every pixel in the filter window. The number of weights depends on the filter window size. Here, 24 weights have to be computed for the filtering of one image pixel. The filter window is shifted first along the input lines representing the image rows, moving one row down every time to the precedent row has been filtered.

Fig. 4: Principle of the input data retrieval for the image filtering
C. Photometric Component:
The parallel calculation of 24 weights in the photometric filter component and the subsequent weighting in the geometric component combined with the final normalization at the filter output require a large amount of resources considering the sparse time of just one pixel cycle. Due to the flexibility of the clock management in FPGAs, this challenge to be accepted.

D. 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.

E. Normalization:
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 guided filter.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Systolic array architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing type</td>
<td>Parallel and pipeline network</td>
</tr>
<tr>
<td>No of pipeline stage</td>
<td>9</td>
</tr>
<tr>
<td>Resources usage</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 1: The specification of proposed architectures for Modified Guided Filter

![Systolic array architecture](image)

VI. RESULT AND DISCUSSION

Table 2: The PSNR comparison for Modified guided filter

<table>
<thead>
<tr>
<th>Input image</th>
<th>Bilateral filter</th>
<th>Guided filter</th>
<th>WLS filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rose</td>
<td>18.0102</td>
<td>18.1490</td>
<td>18.1884</td>
</tr>
</tbody>
</table>

Table 3: The synthesis result for Modified guided filter

<table>
<thead>
<tr>
<th>Architectur e</th>
<th>Bilateral Filter</th>
<th>Guided Filter</th>
<th>WLS Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA Device</td>
<td>EP3CSF256C 6</td>
<td>EP3CSF256C 6</td>
<td>EP3CSF256C 6</td>
</tr>
<tr>
<td>Total logic element</td>
<td>690</td>
<td>463</td>
<td>432</td>
</tr>
<tr>
<td>Total logic register</td>
<td>533</td>
<td>330</td>
<td>281</td>
</tr>
<tr>
<td>Multiplier element</td>
<td>36</td>
<td>28</td>
<td>16</td>
</tr>
</tbody>
</table>

VII. CONCLUSION
In this paper a systolic array architecture for modified guided filter. Compared with bilateral filter, our modified guided filter provides better performance in noise reduction and edge preservation. The maximum operation frequency for systolic array architecture is 146.43 MHz, which is sufficient for HDTV real-time processing.

VIII. FUTURE WORK
Future work would be extended to the 30 frames full HD video capability of process to improve in hardware implementation.

REFERENCES


