2012 4th International Conference on Signal Processing Systems (ICSPS 2012)
IPCSIT vol. 58 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V58.5

Reconfigurable Frame Work for Video Image Enhancement

Ravi Kumar Chadalavada[1+] and Srivatsa S. K[2].

[1] Department of ECE, Prakasam Engineering College Kandukur. A. P, India
[2] Department of ECE, Anna University, Chennai, India

Abstract. This paper presents a method to improve the processing speed of video enhancement using FPGA-based hardware. Video enhancement is one of the most important and difficult components in video research. The aim of video enhancement is to improve the visual appearance of the video, or to provide a "better" transform representation for future automated video processing, such as analysis, detection, segmentation, recognition, surveillance, traffic, criminal justice systems. The development of video processing techniques is closely coupled to the video technology. With the advent of digital technology, the video signal can be digitalized into pixels and stored in a memory, which allows easy and flexible fetch and operation on the pixels to achieve more advanced video processing The VHDL hardware description language has been used in order to make possible a top-down design methodology. Real tine image processing in an application environment needs a set of low cost implementations of various algorithms. This paper presents an adaptive median filter based on FPGA and working at video rate. A video enhancement processor concept is proposed that enables efficient hardware implementation of enhancement procedures and hardware software co-design to achieve high performance, low-cost solutions. The processor runs on an FPGA prototyping board.

Keywords: FPGA, VHDL, median filter, video enhancement.

1. Introduction

The digital video signal contains more dimensions of data than other types of signal such as audio. To enable real-time processing, it requires much more processing power to cope with the ever increasing demand for better picture quality, such as higher resolution and frame rate. To design real-time adaptive and reusable image enhancement architecture for video signals based on a statistical processing of the video sequence. Video processing systems like this one require specific simulation tools in order to reduce the development time. Most of the video and image signals are affected by noise. Noise can be random or white noise with no coherence, or coherent noise introduced by the capturing device's mechanism or processing algorithms. The major type of noise by which most of the signals are corrupted is salt and pepper noise[1]. In past years, linear filters became the most popular filters in image signal processing. However, there exist many areas in which the nonlinear filters provide significantly better results. The advantage of nonlinear filters lies in their ability to preserve edges and suppress the noise without loss of details[6]. Traditionally, the impulse noise is removed by an adaptive median filter which is the most popular nonlinear filter. Its hardware implementation is straightforward and does not require many resources. However, the standard median filter gives a poor performance for images corrupted by impulse noise with higher intensity. A simple median utilizing 3×3 or 5×5-pixelwindow is sufficient only when the noise intensity is less than approx. 10-20%. The electrical system can introduce video noise into a video stream during transmission in any channel. Adaptive median filter provides a simple and effective noise-filtering process. The median value of all the pixels in a population that is, a selected neighborhood block determines each video pixel. The median value of a population is that value in which one-half of the population have smaller values than the

[+] Corresponding author. Tel.: + 91-9849265251.
E-mail address: ravi_ece99@yahoo.com.

median and the other half has larger values than the median value. This paper deals with a highly pipeline hardware realization of the adaptive median filter which is optimized for high throughput. Proposed solution is implemented in an FPGA and its performance is compared against various filters on a set of test images. The goal is to provide a filter suitable for a high performance real-time processing of images corrupted by shot noise of various intensities.

1.1. Video image processing on FPGAs

Application-specific integrated circuit (ASICs) is the first hardware platform for video processing. It is designed for a specific purpose and the design can more be easily optimized so that it usually provides better total functionality and performance. As the applications become more complex, the ASIC design or change takes longer time and the percentage of first time right design decreases. This leads to a higher implementation cost. Therefore, a programmable hardware platform which allows late software modification starts to be used for video processing. Image and Video Processing on embedded devices is a growing trend in the industry today where security is depended on cameras placed everywhere, replacing people behind monitors. FPGAs are preferred for their parallel pixel processing power over sequential microprocessors. Newer FPGAs are packing more gates and requiring lower power, which is certainly attractive features for embedded designers. Instead of a trial-run on an expensive ASIC fabrication process of a custom design, FPGAs offer a cost effective alternative, or at least a prototype before millions of dollars are invested and sometimes to find out the ASIC doesn't perform as expected from simulation on a computer with software.

1.2. The FPGA system

In the past, dedicated DSP chips were an option, but even with high internal clock frequencies, massive I/O capacity and advanced graphics accelerators these are no longer adequate, because they are limited to a certain number of operations per clock frequency. Dedicated graphics processors provide higher performance and are more specialized, though they still have the same limitations. Modern FPGAs do not suffer from such limitations. The capacity has been greatly increased, all of the most recent high-speed interfaces to memory etc. are fully supported, and the number of parallel operations is limited only by the total capacity of the FPGA. Even with lower internal clock frequencies versus dedicated CPUs, this is compensated for by the massive parallelization that is achievable. The most recent FPGAs with dedicated DSP chips are ideal for this type of project, and they function as the heart of the application. Most current DSP algorithms for video processing are fundamentally composed of multiply-accumulate operations (MACs) that are carried out in both dimensions according to the desired resolution. The DSP chips implement the MACs directly at the desired word width and high clock frequency. For applications with fixed processing paths, the separate processing units are connected together in a streaming architecture, where the processing time is the same for all the units in the processing chain and the intermediate results can be buffered internally in the FPGA.

2. Adaptive Median Filter

In video image processing, images are often corrupted by positive and negative impulses stemming from decoding errors or noisy channels. Both are easily detected by the eye and degrade the image qualit. The nonlinear mean filter cannot remove such positive and negative impulses simultaneously. The adaptive median filter performs quite well, but it falters when the probability of impulse noise occurrence becomes high. To overcome this situation, we propose a adaptive median filters with variable window size. The adaptive median filter (AMF) can be defined in several ways. We will use the definition based on the order statistic. In this sense, AMF can be considered as iterative order statistic filter. The iterative processing was introduced in order to detect and replace corrupted pixels only. In eac iteration, filtering windows of different sizes are utilized. The Adaptive Median Filter is designed to eliminate the problems faced with the standard median filter. The basic difference between the two filters is that, in the Adaptive Median Filter, the size of the window surrounding each pixel is variable. This variation depends on the median of the pixels in the present window. If the median value is an impulse, then the size of the window is expanded. Otherwise, further processing is done on the part of the image within the current window specifications. ‘Processing’ the image basically entails the following: The center pixel of the window is evaluated to verify whether it is an impulse or not. If it is an impulse, then the new value of that pixel in the filtered image will be the median value of the pixels in that window. If, however, the center pixel is not an impulse, then the value of the center pixel is retained in the filtered image. Thus, unless the pixel being considered is an impulse, the gray-scale value of the pixel in the filtered image is the same as that of the input image. Thus, the Adaptive Median Filter solves the dual purpose of removing the impulse noise from the image and reducing distortion in the image [2]. Adaptive Median Filtering can handle the filtering operation of an image corrupted with impulse noise of probability greater than 0.2.This filter also smoothens out other types of noise, thus, giving a much better output image than the standard median filter. Adaptive median filters work correctly also for lower noise intensities because they try to use the smallest possible window and so modify only corrupted pixels. The size of filtering window influences the quality of filtering when noise is of more than 40% intensity.

Very diverse FPGA-based custom-computing boards are appearing in the market. These boards possess different interfaces for their communication with the host. But in general, boards devoted to real-time image processing have a USB interface, because it gives them the necessary speed to work as coprocessors. Also, USB bus has a growing popularity due to its interesting properties The fact that we have a 32-bit data bus has a very large influence in the necessary hardware architecture for implementing image processing operations, because it causes that in each read/write operation we obtain/send four image pixels (supposing 8-bit pixels). We have gained benefit from this situation replicating the functional units in order to apply the median filter simultaneously on four pixel neighborhoods [3]. In this way we take advantage of the inherent neighborhood parallelism, and we accelerate the operation four times [4]. Figure 1 presents the approach followed for the simultaneous computation of these four output pixels. Images are divided in pixels (squares) that are grouped in 32-bit words (4 pixels).

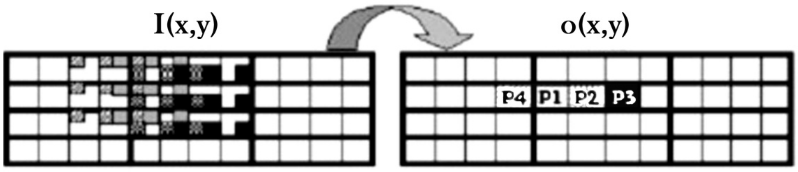

Fig. 1: Simultaneous computation of output pixels.

The value of each output pixel O(x,y) is computed using the 9 pixels of the image I that are inside the 3x 3 mask with centre in I(x,y). Each mask application has been represented with a different texture. Note that the pixel P4 of the previous word is computed and not that of the current word. In this way, it is only necessary to read six words in the input image instead of nine, reducing the number of read operations, and therefore increasing the performance. Pipelining this approach using two stages it is possible to get an architecture that writes four pixels (one word) in the output image in each clock cycle, only reading three input image words by cycle.

3. Sliding Window Architecture

In order to implement a moving window system in VHDL, a design was devised that took advantage of certain features of FPGAs. Fig. 2, 3 shows a graphic representation of the FIFO and flip -flop architecture used for this design for a given output pixel window. FPGAs generally handle flip -flops quite easily, but instantiation of memory on chip is more difficult. Still, compared with the other option, off-chip memory, the choice using on chip memory was clear.

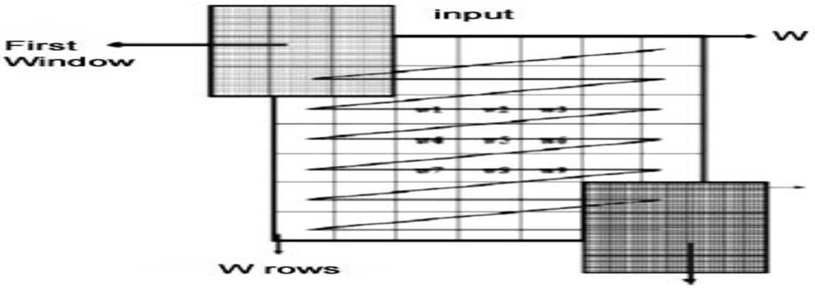

Fig. 2: Sliding window architecture.

Since it was deemed necessary to achieve maximum performance in a relatively small space, FIFO Units specific to the target FPGA were used. Importantly though, to the algorithms using the window generation architecture, the output of the window generation units is exactly the same [5][8]. This useful feature allows algorithm interchangeability between the two architectures, which helped significantly, cut down algorithm development time. A window size was chosen because it was small enough to be easily fit onto the target FPGAs and is considered large enough to be effective for most commonly used image sizes. With larger window sizes, more FIFOs and flip -flops must be used, which increases the FPGA resources used significantly.

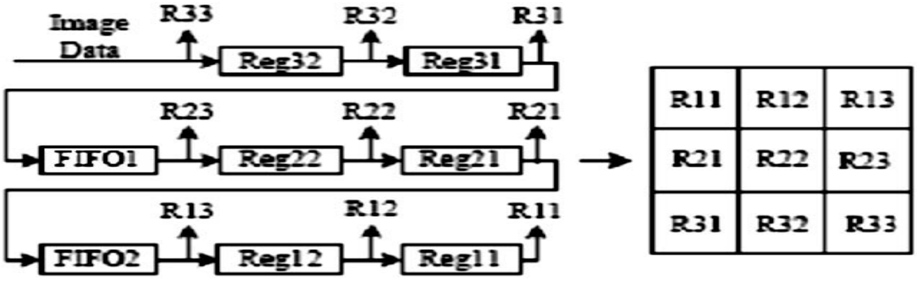

Fig. 3: Reading pixels from window.

3.1. Hardware implementation on a FPGA based video board

To assess the effectiveness of our skeleton-based approach, we have implemented our system on a FPGA based video processing board. A functional block diagram of the FPGA board is given by figure as shown in Fig. 4.

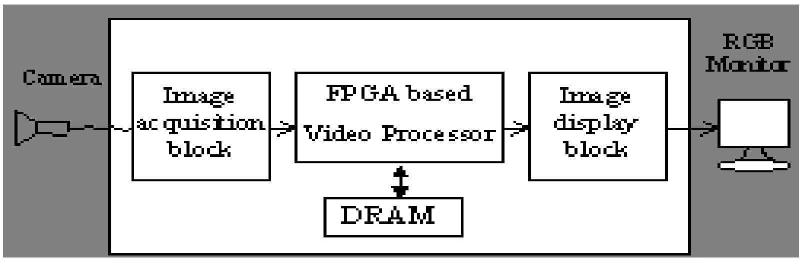

Fig. 4: A functional block diagram of the FPGA board

Bit parallel arithmetic has been chosen to implement the IP operations on the onboard FPGA. This choice is motivated by the fact that bit parallel architectures often lead to a better time-hardware product than bit serial ones. This is mainly due to the existence of dedicated fast carry logic on Xilinx FPGAs. However, in the context of processing real time video, the FPGA board influences the choice of the arithmetic. If bit serial arithmetic is to be used, there is a need to generate a bit clock from the pixel clock. The bit clock frequency is ‘N’ times the pixels’ clock (for an ‘N’-bit pixel). This implies a bit clock frequency of 108 MHz for 8-bit length pixel processing, and 216 MHz for 16-bit length pixel processing. Thus the architectures used will be implemented from bit parallel-based skeletons. A parallel implementation is easier to implement and can be efficiently implemented using dedicated fast carry logic [9]. Image processing is usually performed on pictures stored in an image memory. Achieving global transformations, such as a FF1, requires that one faces difficulties in communication with the computation unit (address processing, high data rates). On the other hand, most low—level image processing is performed on a m*n work window involving pixels of n adjacent image lines. If the image is provided on video format (line by line scanning), on—line processing can be performed, if one assumes that n—l lines are buffer zed, whatever the image height. No random access image memory is therefore necessary [10].

4. Result

The median filters as well as adaptive median filters were described in VHDL, simulated using Model Sim and synthe-sized using Xilinx ISE tools to Virtex II Pro XC2vp50-7FPGA. The implementation costs is expressed in terms of slices. Our FPGA contains 23616 slices in total. The adaptive median filter for video enhancement is designed to remove impulsive noise from video. Therefore, our algorithm's performance was first tested with basic salt and pepper noise with a noise density of 0.25. The next test involves processing images that contain impulsive and/or non-impulsive noise. It is well known that the median filter does not provide sufficient smoothening of non-impulsive noise. Therefore, Gaussian and 'salt and pepper' noise were added to the video which was then processed by the algorithm. The Fig a, b show the performance of the adaptive median filter.

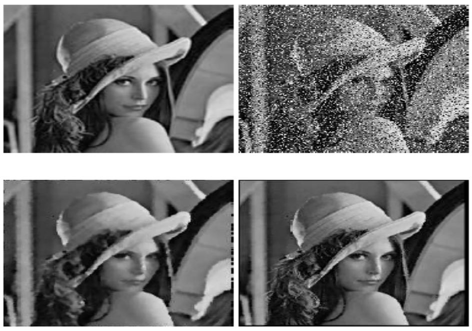

Fig. 4: Results of filtering with a 3X3 median and conditional median filter. From left to right, first row: original Image, noisy image; second row: standard median filter, Adaptive median filter.

5. Conclusion

The architecture is pipelined which processes one pixel per clock cycle, thus to process an image of size *256* x 256 it requires 0.65 ms when a clock of 100 MHz is used and hence is suitable for real time applications The adaptive median filter successfully removes impulsive noise from images. It does a reasonably good job of smoothening images that contain non-impulsive noise. Overall, the performance is as expected and the successful implementation of the adaptive median filter is presented. Specifically, the project requirements include achieving throughput suitable for real-time video, reducing area as needed for implementation in the given FPGA, and producing a noticeable reduction in the artifacts present in the input frame of video.

6. References

[1] Z. Vasicek, L. Sekanina, Novel Hardware Implementation of Adaptive Median Filters, 978-1-4244-2277-7/08/ ©2008 IEEE.

[2] O. Vainio, Y. Neuvo, and S. E. Butner, A Signal Processor for Median-Based Algorithms, *IEEE Transactions on Acoustics, Speech, Processing*, vol. 37. no. 9, September 1989.

[3] V. V. B. Rao and K. S. Rao, A New Algorithm for Real-Time Median Filtering, *IEEE Transactions on Acoustics*, *Speech, Processing*, vol. ASSP-34. no. 6, December 1986.

[4] M. O. Ahmad and D. Sundararajan, Parallel Implementation of a Median Filtering Algorithm, *Int. Symp. on Signals and Systems*, 1988.

[5] D. Tadeusz and M. Szűrők, Mérés és Automatika 37. Évf., 1989. 3.szám

[6] D. R. K. Brownrigg, The weighted median filter, Commun. ACM, 27(8): 807–818, 1984.

[7] S. Marshall, New direct design method for weighted order statistic filters, VISP, 151(1): 1–8, February 2004.

[8] H. Hwang and R. Haddad, Adaptive median filters: new algorithms and results, IP, 4(4): 499–502, April 1995., 37. Évf., 1989. 3.szám

[9] Xilinx Foundation Series Quick Start Guide, 1991-1997. Xilinx. Inc.

[10] W. Zhou and Z. David, Progressive switching median filter for the removal of impulse noise from highly corrupted images, *IEEE TransOn Circuits and Systems: Analog and Digital Signal Processing*, 46(1): 8–80, 1999.

Ch. Ravi Kumar born at Nellore in Andhra Pradesh in India in 1967. He received B.Sc (Physics)degree from Nagarjuna University, in 1985, B.E(Electronics) from Marathwada University in 1991 and ,M.E(Electronics)with specialization in Systems and Signal Processing from Osmania University in 1998.He received Doctorate degree from Bharath University Chennai India in 2011 in field of VLSI. From 1992 on words he is working as Assistant. Associate and Professors levels in reputed Engineering colleges in A.P in India. He is currently working as Principal and Head Department of Electronics in Prakasam Engineering College, Kandukur, Prakasam (Dt)in A.P in India. He is senior member of IACSIT.

Dr. S. K. Srivatsa born in Bangalore in India. Dr.S.K Srivatsa received Ph.D in Engg. From IISC. Bangalore, India. He worked as a Senior Professor In Anna Universit, Chennai, India and he got more than 35 years of teaching experience. He guided and produced more than 32 Doctorates from different universities in India. Presently working as a senior Professor in St.Joseph College of Engineering In Chennai