IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative Analysis of Digital Filters for Image Processing Applications in VLSI

[1]Varsha K, [2]Dr. VijayaLakshmi D,

[1]PG Student,[2]Assistant Professor,

[1]VLSI and Embedded Systems,

[1]Bangalore Institute of Technology, Bengaluru, INDIA

Abstract: Processing of a digital image by the utilization of computer design is normally referred as digital image processing. In image filtering, image processing and computer vision are very useful techniques. It is utilized to remove some unwanted signals which has captured the image and termed as noise. Digital filters are an integral part of signal processing which help us to remove the frequencies which are of less importance or considered unwanted, also known as noise sometimes. Therefore the digital image filtering technique is often used to eliminate noise. Image filters usually follow the software approach in systems. Hardware execution is preferred over software execution for better processing speed. VLSI technology is a good choice for image processing with hardware implementation. The proposed work is with a new hardware accelerator design that filters input data using the Gabor and Box functions and provided to image processing. The approached design will get a great depletion regarding resources compared to other state-of-the-art implementations. In this work, a comparative study of both the gabor filter and box filter on FPGA execution is presented. By using the Xilinx ISE, simulation and synthesis is done. In VLSI, verilog code is a descriptive language and hardware execution in SPARTAN-3 FPGA.

Index Terms - Image Processing,Digital Filters,Gabor Filter,Box Filter.

I. INTRODUCTION

High complexity systems became more and more important in the last years, fundamental role in Deep Learning and Internet-of-Things (IoT) applications. Moreover, in other computationally intensive applications dedicated to multimedia processing, hardware (HW) solutions are usually preferred to software (SW) ones, due to their lower power consumption and period of operation. This is the case, for example, of Visual Search (VS) or object recognition applications. In these kind of applications the filtering stages usually represent the bottleneck of the implemented structures, since they limit the architecture performances in terms of Area-Delay-Power (ADP) product. One of the most used family of functions for image filtering applications is the Gabor one. Gabor filters (GFs) gained more and more importance since Daugman demonstated that they well approximate the behavior of visual cells. Furthermore, global descriptors and features to describe the image could be obtained through GFs responses. These features achieve tolerance to noise in the reference image and are robust to scale changes and rotations of the image. Due to these characteristics, features obtained through such a filtering are of great importance in VS, segmentation and object recognition applications, but could also be used in cepstral analysis for voice recognition. During the years SW implementations using GFs have been presented in literature. To overcome the limitations of these designs, principally in terms of maximum operating frequency, also HW implementations have been proposed, mainly targeted to Field Programmable Gate Array (FPGA) boards . However, the HW designs proposed in literature, always recur to Gabor-like filters in order to simplify the algorithms and thus the structures to be used.

The use of box filters and integral images traces its roots to the field of computer graphics where they were developed for texture mapping (under the name of summed area tables) by Crow in 1984. The use of box filters was introduced in the computer vision and image processing communities by Viola and Jones in 2001 with their landmark face detection algorithm. More recently, several works have shown that box filters and integral images can be used to greatly speed up various computing intensive operations in computer vision and image processing. Bay, Tuytelaars, and Van Gool have used box filters to speed up the detection of the SURF image features. Zhu et al. have used box filters to speed up the construction of Histograms of Oriented Gradient (HoG) features for fast human detection. Tuzel, Porikli, and Meer have used box filters to speed up the computation of Region Covariance descriptors for object detection and texture classification.

II. LITERATURE SURVEY

- Sarwar, Syed Shakib, Priyadarshini Panda, and Kaushik Roy [1] usage of Convolutional Neural Networks (CNN) is as a rule progressively utilized in PC vision for a wide scope of grouping and acknowledgment issues. In any case, preparing these huge systems requests high computational time and vitality prerequisites consequently; their vitality proficient execution is of extraordinary intrigue. In this paper, we diminish the preparation multifaceted nature of CNNs by supplanting certain weight pieces of a CNN with Gabor channels.
- The manuscript recognition suggested by Assoni, Muhammad O, Sabri A. Mahmood [2] is challenging because of the great diversity in human writings. Improving representation, which depends on the visual of the handwritten text, leads to enhanced recognition. In processing, we combine two dominant features based on vision to create strong statistics for handwritten text. Handwritten Text Image Filter is a set of Gabor filters of various scales and trends to capture local features based on format. Gabor filter response features are arranged in two layouts. Provides bag-off features for learning statistical Gabor features and Gabor descriptions, as well as robust statistics for handwritten text.
- Adak, Chandranath [3] introduced the Gabor filter and an effective edge detection method that supports uneven clustering. The input image is facilitated by the Gabor function, so a rigorous clustering concept is used for specialization in marking the edge by a smooth calculation method. Hysteresis thresholding is used to push the original production, i.e. the ends of the input image. To highlight the effect, specific technology was compared to other edge detection methods.
- Chadri, Kunal Narayanan, Aret Munos Barutia, and Michael Anser [6] all believed that an image could be filtered in different sizes, lengths, and windows at a fixed cost per pixel. In this method an acceptable global pre-integrator is applied and then a degree-wise custom localization mesh is applied. We demonstrate the required theory for a 1D case that uses the B-spline formality, so it can be extended to 2D as appropriate using radial-uniform box splines. The dimensions and ellipse of the radial-uniform box splines are adjusted accordingly. Also, as the order grows, they merge with the Gaussians. Finally, we demonstrate a quick and practical directional filtering algorithm that matches local image properties.
- Eustace Paincretel. [8] Gaborzet introduced the VLSI architecture for calculating facial feature from video having faces for PC focused applications. Based on multiplier-cache units with built-in ROMs, the specific structure maps to the network of most pipeline processing components (PEs). It is compatible with algorithm-specific ASIIC or Gabor filtering, most widely used removal techniques for face processing application as face tracking and face recognition.
- Kaiming He et.al [11] suggested a type of clear image filtering - the guided filter. The guided filters, taken for the linear model, produces a filtering output based on the contents of the guided image, which may be an input image or noise affected image. The guided filter act as edge-protecting sensitive worker like the accepted bilateral filter, but have a good performance near the edges. The theoretical relationship with the Laplacean matrix, so it is a large general idea than the sensitive worker, and can build superior use of the structures in the guiding image.

III. PROPOSED METHODOLOGY

1. GABOR FILTER

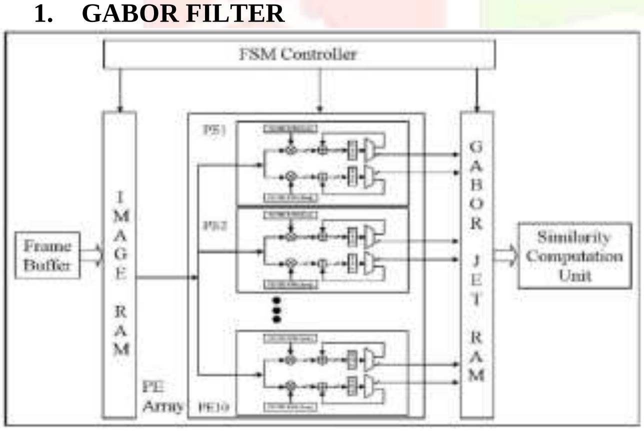

Fig 1.1. Block diagram of the proposed Gabor Filter

In image processing, a linear filter is utilized to identify the edge of the gabor filter named subsequent after Dennis Gabor. The frequency and direction illustration of gabor filters is comparable to that of the human visual system and to establishing texture illustration with intolerance. In spatial domain, the 2D Gabor filter is a Gaussian kernel purpose that modulates the sinusoidal plain signal. Gabor filter funcions on normal cells in the visual cortex of the human brain. So, image study with gabor filters is probable visualized in the human visual system.

The gabor Filter blocks are explained briefly:

1. **Image RAM:** In 16 x 16 pixel image, the range of filter mask is stored in the internal image RAM for initial calculation. By using 8 bit gray level pixel values, all calculations are done. Image RAM of 16 pixels should be reorganized with a row / column for the next gabor jet calculations for the subsequent pixel in the specified FFP's area.
2. **Gabor PE:** Figure 3.2 shows the essential processing component of the Gabor filter bank for Gabor jet calculation. Each PE contains (i) two real and imaginary hazard multipliers in each PE (ii) pixel intensity is received two orders, (iii) two orders with output registers, summation and storage of each 12 bit wide, (iii) size 246 x 246 bs (26600 b). Each PE calculates and collects 256 in-

between values, and assigns the gabor coefficient to 1 pixel and 1 Gabor filter at the finish of 256 clock cycles earlier than manipulating the gabor coefficient of the similar pixel for the subsequent gabor filter.

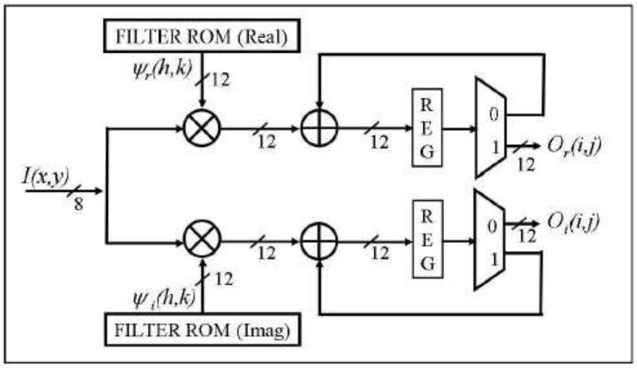

Fig.1.2.Gabor processing element

3. **Gabor Filter Masks:** It as to be initially tested in Matlab to get accurate floating point gabor modules. The filter masks (40) have composite coefficients that can be abbreviated to set of numbers with an accuracy of 12 bits, to get normalization and mathematical accuracy.
4. **Gabor filter ROM:** Every PE has an integral part. Every ROM 4 16 x 16 pixel Gabor filter stores 4 multipliers that fit the original and inary components of the mask. Although shown as separate ROMs in Figure 3, the original and inary components of the filter coefficient are kept in a one ROM for quick verification.
5. **Input Image Pixel Access:** In our structure, 10 Gabor filter is equipped with 10 PEs to count fractions in parallel, each using the same pixel at the same time. This ensures that 256 pixels per 10 files are read continuously at once. For the next 10 sets of filters, this input image re-reads the pixels, thus calculating the Gabor jet for 1 pixel, requiring 4 read cycles for every 40 filters. The option above 10 limits the fan to 10 from each pixel.
6. **Gabor Jet Memory Write:** The 10 Gabor jet elements computed every 225 cycles are written into the Gabor jet RAM for further processing by the similarity computation unit. For each pixel in the search area of a FFP, the Gabor jet RAM requires 40 (Gabor jet elements) x 12 (bits/jet)x 2 (real and imaginary). For n x n search area around a FFP, we compute n x n Gabor jets. This RAM can be optimized when coupled with the pipelined similarity computation unit.
7. **Input and Output sequencing:** The finite state machine (FSM) organizer monitors the correct input and output series and time collection of results. It reads 256 pixels, which is enough for one FFP and reads from the buffer and stores in the image RAM. Gabor jet module 10 process this data for gabor jet filter process, writing its output to the gabor jet RAM at the ending of each 256 clock cycle. This way, a total of 40 filters can be rated for a total of 900 clocks using single pixel values. For the subsequently pixel in FFP area, in image RAM only the n pixels needed to modernized and subsequent gabor jet is counted pending all n x n pixels can be also counted. The calculation of unit, in the gabor jets after that FFP as written above and note them into the gabor jet RAM for similar calculation unit consumption.

2. BOX FILTER

Box filtering is mainly an average-round-pixel kind of image filtering. This is a usually used mathematical function for image filtering. A convolution filters offer a way to multiply two arrays to build a new array (third). In Box Filtering, to get the filtering result the image pattern and the filter kernel multipled. The filter kernel describes how filtering occurs, which essentially determine the type of filter. The box filtering lies in the ability to write simple image filters that can do sharpening, embossing, edge-detection, smoothing, motion-blur and more, using the appropriate filter kernel provided.

Temporary filters uses multiple versions of each pixel value which are used in the same place at different "times", the spatial filter uses the same "time" pixels, but at different spatial latitudes in the image. When a static scene is temporarily filtered, it is assumed that the pixels have the same value from the same place and at different times. When only the extreme image is available, a spatial filter is used. A easy spatial filter is an average filter that returns a pixel with its average and native neighboring pixels. A box filter with a weight equal to 9 pixels in a 3×3 window is the most common average filter. This option increases the weight when it comes to additive noise. Box filtering is similar to a temporary filter, where samples from adjacent pixel locations are used approximately for multiple samples of pixels in a single location.

3. Advantages:

- Exclusively designed for FPGAs with distributed and block RAMs
- Less memory usage
- High processing speed
- Low computational complexity.

4. Applications:

Developing fields of digital image processing technology such as

- Fingerprint authentication system
- Facial recognition
- Iris identification
- Speaker recognition
- Digital watermarking
- Medical Image Processing like X-rays
- Machine Industrial Vision Applications
- Image processing tool for remote sensing
- Image processing for remote sensitive data
- Artificial neural networks in image processing

IV. HARDWARE AND SOFTWARE REQUIRMENTS

1. XILINX ISE

Xilinx ISE (**I**ntegrated **S**oftware **E**nvironment) is a software tool developed by means of Xilinx for the integration as well as examine of HDL designs, and design and analysis to integrate (analyze and analyze) their designs. Organize the objective device with the programmer. Software tool shaped by Xilinx for study of HDL designs and synthesis, enable the developer to compile(synthesis) their designs, execute timing analysis, inspect RTL diagrams, simulate a design's to stimuli, and organize the main device with the programmer.

The Xilinx delivers the Xilinx ISE (Integrated Synthesis Environment) is a creation apparatus meant for summation and investigations of HDL structures, providing the workspace for the designer to orchestrate ("arrange") their ideas, compute timing investigation, view at RTL outlines, reenact a ideas reply to a variety of upgrades, and place the objective tool by means of the software engineer.

Xilinx ISE is a structure orbit for device product from Xilinx, and is strongly joined to the design of chips, and it does'not entertain the Chips designed by different vendors . The Xilinx ISE is essentially employed meant for design union and plan. The design is tested at the framework level using ISIM or Model Sim. The other applications of the Xilinx ISE includes the Embedded development kit (EDK) , Software development kit (SDK) and Chip Scope Project.

2. MATLAB Tool

MATLAB(matrix research center) is a worldwide used numerical computation condition and finite designing language created by MathWorks. It allows web controls, graphing of capacity and information, processing calculations, making of UIs. The C , C++ , C#, Java , Fortran and Python is the dialects used for interfacing with projects.

Many designers and researchers use MATLAB to breakdown and plan the software and work that changes our reality. The designing Process involves imagination and choice of the bits of knowledge from information. The most approach to convey the information of computational mathematic expression is based on lattice based MATLAB language. It allows investigation, Experimentation and revelation. These works is fully tried and tested to obtain the best yields.

MATLAB provide work area to process our thought or idea. We can run our designs on bigger computations indexes, and raised to bunches and mists. The software code can be used along with various languages, providing opportunity to send computations and applications inside web, venture, and generation frameworks.

The image which we have considered for the proposed work for both box and gabor filter will be given as input to the Matlab. The output values obtained in Matlab are given as input to both gabor and box filters in Xilinx.

V. RESULTS

The implemented system is implemented using Xilinx. The simulation results of gabor filter is obtained and as shown in the figure .5.1 below.

Fig.5.1 Simulation results of gabor filter

The simulation results of box filter is obtained and as shown in the figure .5.2 below.

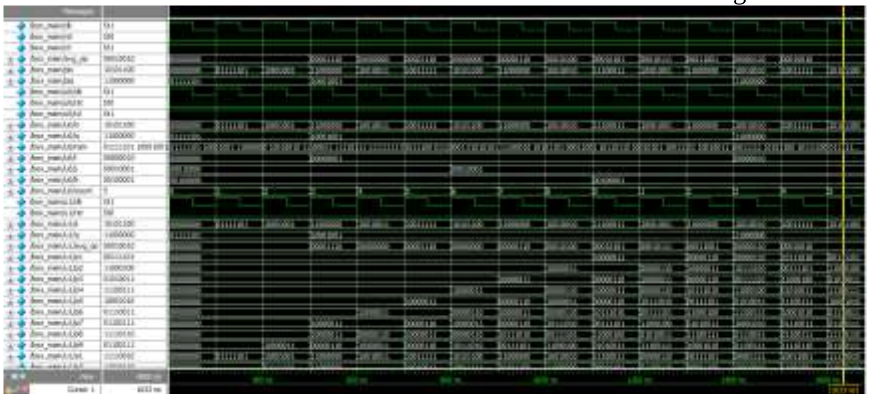

Fig.5.2 Simulation results of box filter

Table I shows the comparative study of area, power and delay of both Gabor Filter and Box Filter.

TABLE I: Comparison of Resources of Gabor Filter and Box Filter

	Area	Power(mW)	Delay(ns)
Gabor Filter	LUTs- 279	391	13.414
Box Filter	LUTs-460	178	4.855

It can be observed from the Table I that Box filter performs well in terms of Power and delay compare to Gabor Filter. The quality of the output image after box filtering is good compared to Gabor filter. The Gabor filter suffers with poor image quality after filtering.

Fig 5.3 and Fig.5.4 below shows the example of an image obtained using Gabor and Box Filter in Matlab.

Fig 5.3. Gabor filter using Matlab

Fig 5.4 Box filter using Matlab

VI. CONCLUSION AND FUTURE SCOPE

1. Conclusion

According to present technology, the future depends entirely on the virtual world. Now the whole thing is based on real data relocate, we need a voiceless system for direct data transfer as we know it. Most image filtering methods can be implemented efficiently with a minimum number of functions per pixel. In this work we have compared the area, power and delay parameters for gabor and box filters. Improved design approach for Gabor filter and box filter implemented with Xilinx. Image processing is fast and inexpensive. The area of the design is considerably lower when the performance of the filters is well maintained. According to this work, we preformed a comparative analysis of gabor and box filters, which maintains the time complexity and quality of the output images. By comparing different filtering methods it can be concluded that the box filter is superior to the Gabor filter. It produces better output and effectively reduces noise by making the image smoother and sharper. This work was successfully tested using the Spartan-3 FPGA kit.

2. Future Scope

Even though box filters produce good image quality, steps must be taken to reduce the LUT size in the Box Filter. In future, box filter can further be enhanced to be used along with other digital image filters namely gradient and guided Image filter in order to achieve reduced area.

References

[1] Sarwar, Syed Shakib, Priyadarshini Panda, and Kaushik Roy. "Gabor filter assisted energy efficient fast learning convolutional neural networks." In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1-6. IEEE, 2017.

[2] Assayony, Mohammed O., and Sabri A. Mahmoud. "Integration of Gabor Features with Bag-of-Features Framework for Arabic Handwritten Word Recognition." In *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, pp. 1-4. IEEE, 2017..

[3] Adak, Chandranath. "Gabor filter and rough clustering based edge detection." In *2013 International Conference on Human Computer Interactions (ICHCI)*, pp. 1-5. IEEE, 2013.

[4] Gong, Yuanhao, Bozhi Liu, Xianxu Hou, and Guoping Qiu. "Sub-window box filter." In *2018 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1-4. IEEE, 2018.

[5] Pires, Bernardo Rodrigues, Karanhaar Singh, and José MF Moura. "Approximating image filters with box filters." In *2011 18th IEEE International Conference on Image Processing*, pp. 85-88. IEEE, 2011.

[6] Chaudhury, Kunal Narayan, Arrate Munoz Barrutia, and Michael Unser. "Fast adaptive elliptical filtering using box splines." In *2008 15th IEEE International Conference on Image Processing*, pp. 785-788. IEEE, 2008.

[7] Lin, Meng-Chun, and Lan-Rong Dung. "On VLSI design of rank-order filtering using DCRAM architecture." *Integration* 41, no. 2 (2008): 193-209.

[8] Painkras, Eustace, and Charayaphan Charoensak. "A vlsi architecture for gabor filtering in face processing applications." In *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 437-440. IEEE, 2005.

[9] Chen, Jian, Yoshinobu Sato, and Shinichi Tamura. "Orientation space filtering for multiple orientation line segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 5 (2000): 417-429.

[10] Paris, Sylvain, and Frédo Durand. "A fast approximation of the bilateral filter using a signal processing approach." In *European conference on computer vision*, pp. 568-580. Springer, Berlin, Heidelberg, 2006.

[11] He, Kaiming, Jian Sun, and Xiaoou Tang. "Guided image filtering." In *European conference on computer vision*, pp. 1-14. Springer, Berlin, Heidelberg, 2010.

[12] Tomasi, Carlo, and Roberto Manduchi. "Bilateral filtering for gray and color images." In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pp. 839-846. IEEE, 1998.

[13] Kywe, W., D. Fujiwara, and K. Murakami. "An Approach to Linear Spatial Filtering Method based on Anytime Algorithm for Real-time Image Processing." In *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, no. 12. 2012.

[14] Li, Zhengguo, Jinghong Zheng, Zijian Zhu, Wei Yao, and Shiqian Wu. "Weighted guided image filtering." *IEEE Transactions on Image processing* 24, no. 1 (2014): 120-129.