Design of Real-time SIFT Feature Extraction

Ruiqi Wang
Department of Electronics, Electrical and Computer Engineering
SungKyunkwan University
Suwon, Republic of Korea
wangrqrqwang@163.com

Jae Wook Jeon
Department of Electronics, Electrical and Computer Engineering
SungKyunkwan University
Suwon, Republic of Korea
jwjeon@yurim.skku.ac.kr

***Abstract*—A real-time hardware architecture based on scale-invariant feature transform algorithm (SIFT) feature extraction with parallel technology has been introduced in this paper. The proposed parallel hardware architecture could be able to extract feature via a Field-Programmable Gate Array (FPGA) chip efficiently, which provided the real-time performance and the similar accuracy with software implementation. In terms of hardware resource consumption and speed, the original SIFT algorithm has been significantly optimized in the following aspects: 1) Down-sampling has used to replace with up-sampling for purpose of saving the interpolation calculation. Besides, the flexible Gaussian blur value and self-circulation scale space are proposed to simplify the key point detection operation. 2) Optimized key point detection method replaces the original key point detection methods for there won't be other key points in the neighbor 8 pixels in the same scale if a certain pixel were defined as a key point. Compared with the SIFT algorithm implemented in the software, this kind of architecture based on FPGA performance is similar with software methods and realizes the efficient and real-time.**

Keywords — Scale invariant feature transform (SIFT); Field-Programmable Gate Array (FPGA); feature extraction; real-time; parallel architecture

I. INTRODUCTION

Feature extraction now is an important component of computer vision fields. Computer vision, which has evolved from "describe what it saw" in the late 1960s to an interdisciplinary field that deals with how computers know from digital images or videos nowadays is concerned with extracting information from images. As to feature extraction, it is most widely used in artificial intelligence including machine learning, pattern recognition and in image processing. However, in terms of the complex computationally requirement, it is sometimes not possible to implement their direct application on mobile, robot platforms [1].

However, with the rapidly development of digital integrated circuits and devices, the way to obtain a computer vision system with a low power consumption and acceptable quality on a cheap device is to use a dedicated architecture, there is DSP (Digital Signal Processor), ASIC (Application Specific Integrated Circuit), FPGA [2] and other solutions and products [3]. Usually, with the development of the shorter and cheaper cycle, FPGA becomes more extensive, which has the advantages of low-power consumption and supports a variety of standard I/O resources. In our paper, we have prototyped an FPGA architecture to implement hardware SIFT algorithm and design due to discussed reasons.

There exist several methods in the field of feature extraction, like edge detection, corner detection, blob detection, ridge detection etc. In 1999, David Lowe [4] proposed one of the most robust methods which is called SIFT algorithm. With the advantages of which are invariant to image transition, scale, rotation and partially invariant to illumination changing, three-dimensional rotation and nonlinear geometric distortion, the SIFT algorithm begin to use to extract and describe local features in images. In recent years, there are more robust features come up with, such as Speeded up Robust Features (SURF) [5], Binary Robust Invariant Scalable Key points (BRISK), Binary Robust Independent Elementary Features (BRIEF) etc. Among these, the SIFT is the most robust one.

Sheng Zhong [6] presented an effective new hardware design that the SIFT feature detection step is implemented in a fully parallel hardware based on FPGA, while the SIFT description step in a high-performance fixed-point DSP chip. The combined DSP and FPGA architect is mainly focus on the speed. However, the power consumption and high price of these processors is enormous. Zhang and Chen [7] illustrated some optimization technology and proposed two parallel SIFT algorithms to improve the performance of implementation analysis on multicore which combined with the 8-core system and a 32-core Chip Multiprocessor (CMP) simulator. However, the power consumption of these processors is still enormous as Sheng Zhong's proposal.

As to our proposal, according to the analysis of our result. By feature extraction, the accurate rate is over 80% compared with the software implementation. The feature extraction system has better real-time performance. In addition, the utilization of resources has greatly reduced because of the design with top-down modular structure and pipeline architecture.

II. INTRODUCTHEION OF THE SIFT ALGORITHM

SIFT algorithm can be separated in to two main steps, one is the key point detection and the other is the descriptor generation.

978-1-5386-1127-2/17/$31.00 ©2017 IEEE

A. *Build a Different of Gaussian Pyramid*

The Gaussian blur has been considered mainly as two parts. One is Gaussian kernel window and the other is two-dimension (2D) convolution operation. In computer vision, especially in image processing, Gaussian filter is one of the most important and widely used filtering algorithms. Gaussian filter(G) is defined as:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (1)$$

Where G is the Gaussian kernel at the location with coordinates x and y, σ is the Gaussian parameter. If the value of σ is large, the image smoothing effect will be higher.

In general, smoothing can be effected by convolving the original image I(x, y) of the size h, and w with a Gaussian kernel G(x, y). It is obtained by computing the sum of products among the input image and a smaller Gaussian matrix of the size (3×3). So the result of Gaussian blur is as equation (2). A 2D convolution using a 3×3 kernel and 3×3 input image is illustrated in Figure 1:

$$F(x,y)=\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}G(i,j)I(x-i,y-j) \quad (2)$$

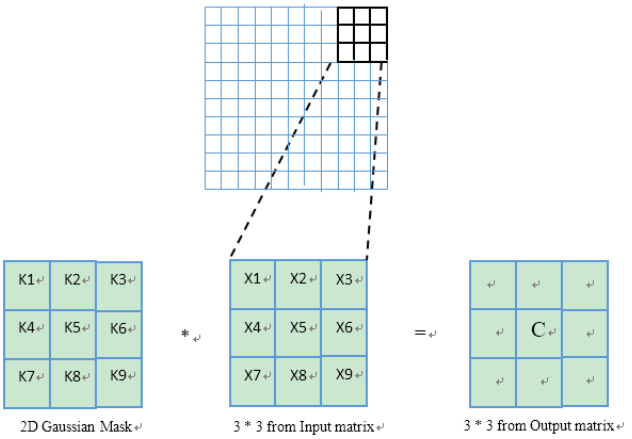

Figure 1：Gaussian convolution operation

It is important for improve the accuracy and complexity in whole FPGA system. So, the choose of octaves and levels in creating a scale space is essential for affecting the accuracy and complexity. According to the MATLAB simulation, there are 2 octaves and 4 levels, which have the better result and decrease the calculation of the algorithm and the resource consumption. It is important to take the original image and generate progressively blurred images. In general, Down-sampling generates several different sizes of images. Through the Gaussian blur first, and then down-sampling will generate another Gaussian scale space. The process to construct the different of Gaussian pyramid (DoG) images is shown in Figure 2. The Gaussian pyramid contains two octaves and four scales per octave as mentioned previously. The second octave is down-sampling from the original image. The DoG space is built up by simply applying the subtraction to two adjacent Gaussian scale images - the above scale and the below scale, pixel by pixel. [8]

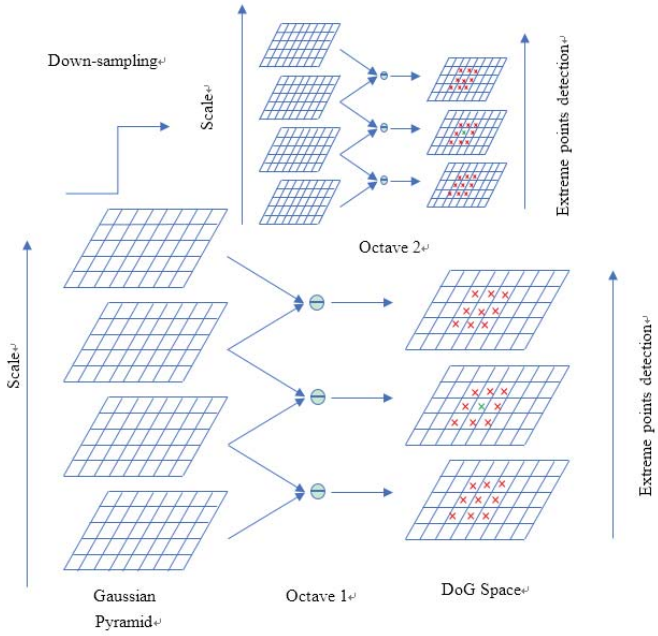

Figure 2: Gaussian DoG Pyramid

B. *Key Point Detection*

As mentioned in the Figure 2. The key points are extracted from the 26 neighbor pixels by finding the local maxima or minima value in the DoG Space. Besides, eight of them are the same scale with the key points and the other 18 neighbors are from the above and below DoG scale space.

In order to making the lowest contrast key point, the edge response key point should be removed in this module for the accuracy. According to the SIFT algorithm which is proposed by Lowe [4], the edge response is proportional to Hessian matrix. Hessian matrix is as follows.

$$H=\begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (3)$$

The elements in matrix is defined in the DoG module in Part A. The value of edge response is calculated with the trace of the Hessian matrix and the determinant.

$$\frac{Tr(H)^2}{Det(H)}=\frac{(\lambda_{\min}+\lambda_{\max})^2}{\lambda_{\min}\lambda_{\max}}=\frac{(1+\sigma)^2}{\sigma} \quad (4)$$

$$\frac{Tr(H)^2}{Det(H)}\le\frac{(1+\sigma)^2}{\sigma} \quad (5)$$

As mentioned in the equation (4) and equation (5), Dxx and Dyy are discrete values which are convenient for realizing

the edge response module. Thus, simple addition and multiplex could be implemented to design the edge response architecture.

C. Calculation of Gradient Magnitude and Direction

For descripting the feature key point, the gradient magnitude and direction is used to describe. The magnitude and direction of the key point is calculated from nigh neighbor pixels around the key point. The equation (6) and equation (7) demonstrated how to compute the gradient magnitude and main direction. [9]

$$\begin{aligned} grad(x,y,\sigma) &= |L(x+1,y) - L(x-1,y)| \\ &+ |L(x,y+1) - L(x,y-1-1)| \end{aligned} \tag{6}$$

$$\theta(x,y,\sigma) = \tan^{-1}\frac{|L(x,y+1) - L(x,y-1-1)|}{|L(x+1,y) - L(x-1,y)|} \tag{7}$$

D. Descriptor Generation

In this part, in terms of the invariance of rotation and illumination, the SIFT descriptor established in to three steps: firstly, the direction of each key point which is selected in 16*16 windows is transferred into eight bins, and then it does the 2D convolution with Gaussian kernel with a scale equal to one half of the descriptor window. Secondly, 4*4 window is obtained from the 16*16 window. Finally, a 16*8=128 elements' vector has been built to descript the main direction. The algorithm illustrates in Figure 3.

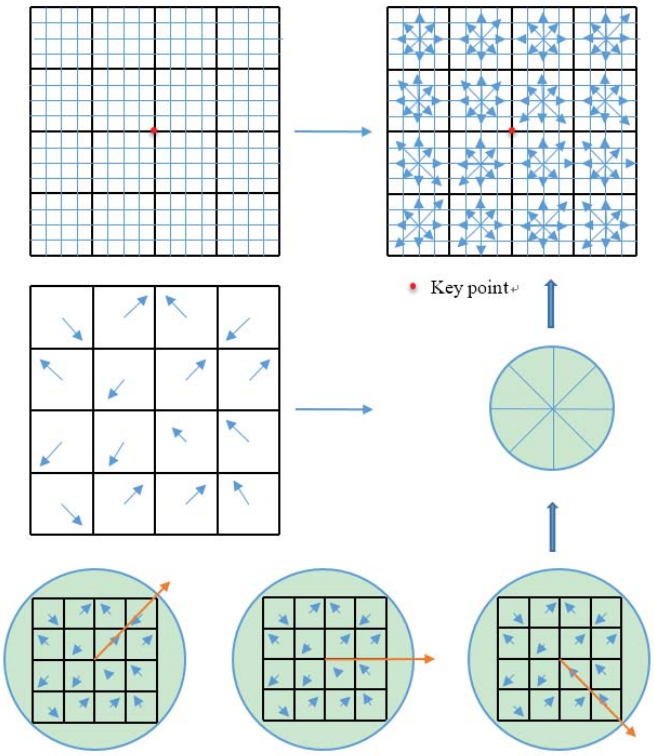

Figure 3: Key point vector calculate

III. Implemetation of The SIFT Algorithm

In this part, it mainly illustrates the parallel architecture of SIFT algorithm. As show in Figure 4, the original image firstly implements in the Gaussian blur module, and then one of data finds the extreme point and the other data finds the gradient magnitude and the angle by using CORDIC algorithm. Combined the two path, histogram will be done and then, the main direction and descriptor will be generated. The result can be send to PC and displayed.

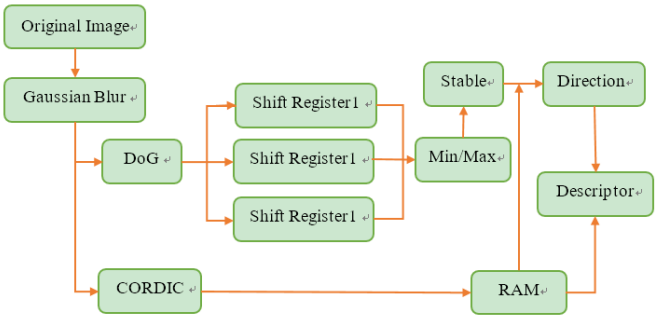

Figure 4: the architecture of the SIFT algorithm

A. Gaussian blur and the DoG implementation

Traditional SIFT algorithm proposed the FIFO for restoring the data. However, in this paper, it proposed the shift register for shifting and restoring the data, which is better for saving FPGA resources. In the Figiure5, firstly, the data which came from the original image could be send to the shift register 1, 2, 3 orderly. Then, the matrix window will be generated for the Gaussian convolution. The Gaussian parameter is stored in the ROM and can set by user freely. Through the simulation, it is proved that for a 256*256 image, 3*3 Gaussian window is better. The image of the first octave is 256*256, and the Down-sampling to 128*128 for the second octaves. Gaussian blur has been done for both octaves and then DoG data has been established.

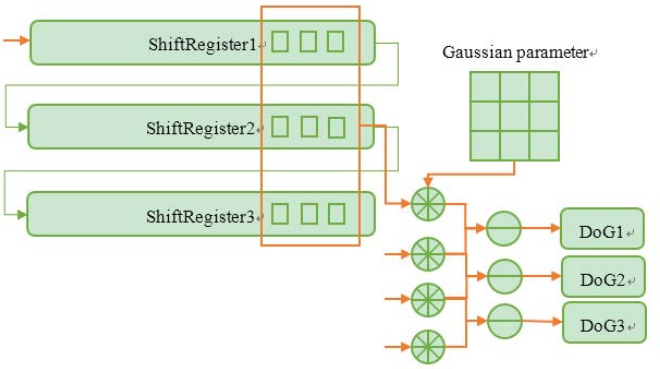

Figure 5: The architecture of Shift Register

B. Key point detection and implementation

When it comes to key point detection, the point which will be detected in the DoG image will be found the maximum or minimum point between the 8 points and the 9*2 neighbor points from the above scale and the below

scale. As we proposed in the previous, if there is a key point, it won't be any other point in these 26 points. So, it won't be detected anymore.

When the key point has been detected, it is necessary to implement the stable key point detection as it mentioned. In terms of getting the low contrast points and strong edge response points, Hessian matrix will be used to remove the edge response key points. Figure 6 demonstrates that the edge response mainly focus on the itself and the 8 neighbors, the 2 times and 4 times multiplex and left, right shift could be implemented to realize it.

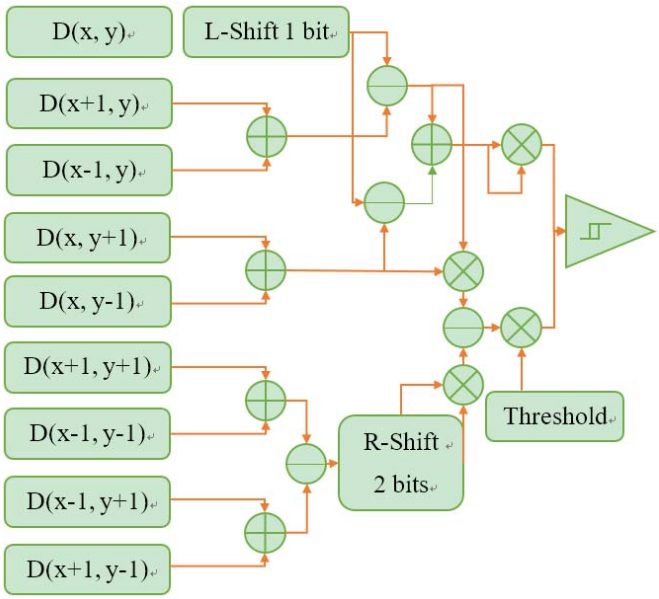

Figure 6: Key point edge response remove module

C. *Main direction calculation*

SIFT algorithm define the main direction of each key point to obtain the rotation invariance. Figure 7 shows how to calculate the main direction of each key points.

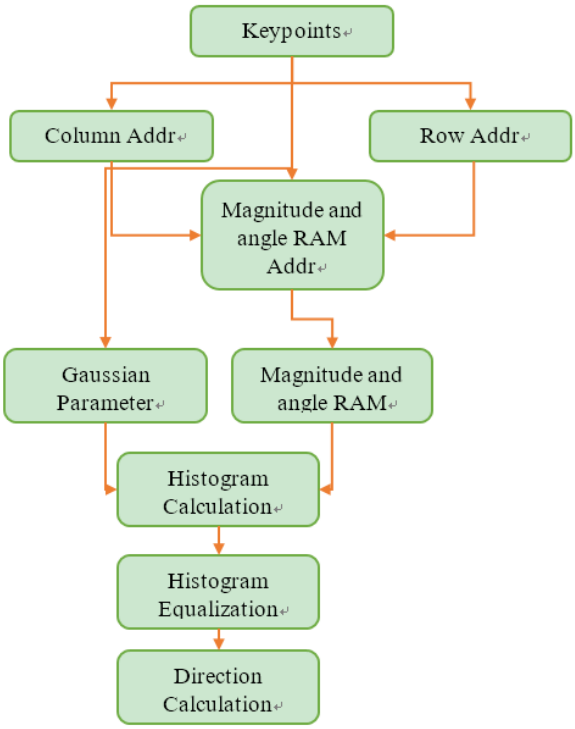

Figure 7: Main direction calculation module

Firstly, every key point will be calculated with the gradient magnitude of the certain neighbor which has been done by the Gaussian blur, with the center of the certain scale location. In this paper, we chose the 1.5*3*σ, and σ is the Gaussian parameter in the scale of the key point. Then, histogram weighting will be calculated in the key point and its neighbors. There are two kinds of weighting parameters, one is the magnitude of the key point and the other is the fixed Gaussian weight function. Gaussian weight kernel is used to strengthen the influence of the magnitude of the neighbor of key points.

Histogram equalization will be established in order to decrease the influence of the discrete key point to the histogram. Finally, we will get 36 histograms, and main direction will be found by comparing the nearby histogram which has the threshold of 0.8. The column and row address will be calculated for locating. The process shows in Figure 8.

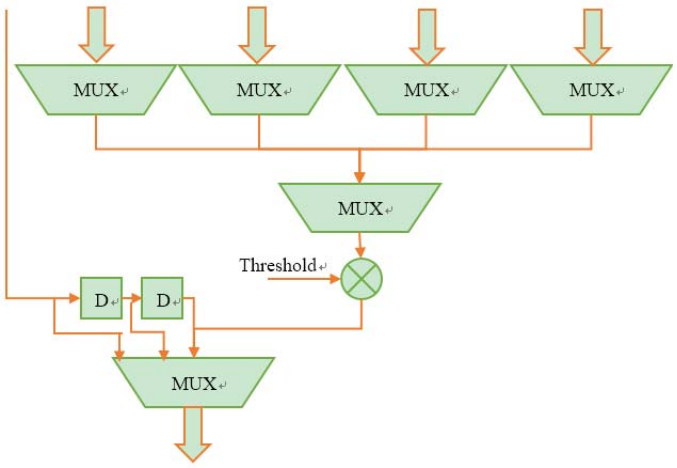

Figure 8: Main direction calculation architecture

D. *Key point Descriptor implementation*

From the calculation of previous part, the location, space scale and the main direction of each key point has been done. According to this information, we could establish the descriptor for descripting the invariance. Typically, Lowe proposed in original algorithm that a descriptor with 128 elements is established for every key point. The vector has the feature neighborhood information with the high uniqueness.

IV. Results

In this paper, the original picture is stored in a ROM which is a IP core in FPGA hardware platform. The process as the paper mentioned is implemented in Xilinx platform, using the Verilog language. Based on that platform, the result shows in the Figure 9 at 30 fps feature extraction. The result is as follows:

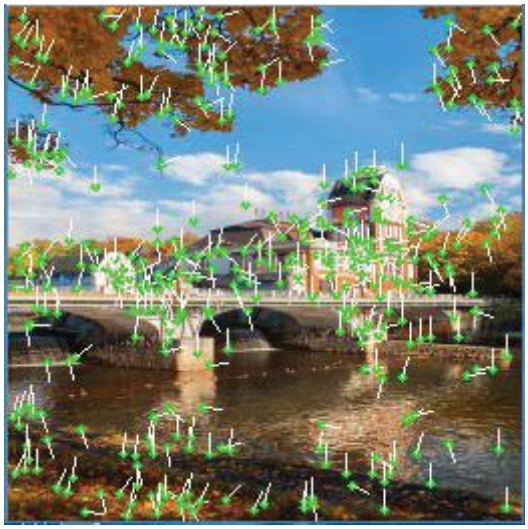

Figure 9: result1 based on FPGA

The SIFT features show in the Figure 9. The size of image is 256*256 and there are totally 958 key points in the picture, and after the edge response remove operation, there are 380 key points left.

V. Acknowledgment

This research was supported by MSIP, Korea, under the G-ITRC support program (IITP-2015-R6812-15-0001) supervised by the IITP, and by Basic Science Research Program through NRF of Korea, funded by MOE (NRF-2010-0020210).

References

[1] V. John, K. John. “Fully pipelined FPGA-based architecture for real-time SIFT extractiona seeing robot rover,” Microprocessors and Microsystems, Stanford, vol.40, pp. 53-73, 2016.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Ronak, B.,Fahmy, S.A. “Mapping for Maximum Performance on FPGA DSP Blocks,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.6, no.1, pp.1-13,2015.

[3] J. Herbert, S. Wilson, “FPGA implementation of a high-speed, real-time, windowed standard deviation filter,” Electronics Letters, vol.34, pp.22–23,2016.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] Se S, Lowe D, Little J. “Vision-based mobile robot localization and mapping using scale-invariant features” [C]. International Conference on Robotics and Automation, 2001,

[5] H. Bay, T. Tuytelaars, and L.V. Gool, “SURF: Speeded up robust features,”ECCV, pp.404–417, 2006.

[6] S. Zhong, J.h. Wang, “A real-time embedded architecture for SIFT,” Journal of Systems Architecture, vol.59, pp. 16–29, 2013.

[7] Q. Zhang and Y. R. Chen, “SIFT implementation and optimization for multi-core systems,” in Proc. IEEE IPDPS, Apr. 2008, pp. 1–8.

[8] Jie Jiang, Xiaoyang Li, Guangjun Zhang. “SIFT hardware implementation for real-time image feature extraction”. IEEE Transactions on Circuits and Systems for Video Technology, VOL.24, No.7, July 2014.

[9] J.Q Peng, Y.H.Liu, C.Y.Lyu, Y.H.Li, W.G.Zhou and K.Fan. “FPGA-based parallel hardware architecture for SIFT algorithm”. IEEE International Conference on Real-time Computing and Robotics. June 6-9,2016.