Proceedings of The 2016 IEEE International
Conference on Real-time Computing and Robotics
June 6-9, 2016, Angkor Wat, Cambodia

FPGA-Based Parallel Hardware Architecture For SIFT Algorithm

J.Q. Peng, Y.H. Liu, *Fellow, IEEE* , C.Y. Lyu, Y.H. Li , W.G. Zhou and K. Fan

***Abstract*—A parallel hardware architecture for real-time image feature detection based on the SIFT (Scale Invariant Feature Transform) algorithm has been presented. The proposed parallel hardware architecture is completely stand-alone; it reads the input data directly from VITA2000 and provides the results via a Field-Programmable Gate Array (FPGA); image key points are extracted in combination with SIFT IP Core. Our proposed parallel hardware system could be able to detect feature points up to 30 frames per second with the resolution 1920*1080; the proposed method has the similar accuracy to PC implementation. Experimental results shown that the hardware architecture for SIFT algorithm realizes fast feature extraction, the shortcomings of massive calculation and low speed in the process of extracting image features have been efficiently improved, which meets the real-time requirements in feature matching system. The achieved system performance is at least two orders of magnitude better than a PC-based solution; this algorithm can be applied to feature matching in the field of Robotics.**

Keywords: **Scale invariant feature transform**; **FPGA**; **feature matching**; **Robotics**

I. INTRODUCTION

With the fast development of computer vision, the image processing algorithms has continuous improvement. In most cases, such as robot navigation, panorama stitching, face recognition, it is necessary to extract image features. However, these algorithms are normally implemented to be executed on PC, due to computationally demanding for feature detection, their direct application on mobile robot platforms sometimes is not possible [1].

Fortunately, with the continuous development of integrated circuits, there has been ASIC, DSP, FPGA [5] and other products, the highly integrated FPGA and EDA technology [6] making the application of FPGA becomes more extensive. The advantages of FPGA are highly configurable FPGA logic, high-speed transceiver, low-power and supports a variety of standard I/0 resources. As a result, we can use it to do complex parallel algorithms. In one sense, hardware acceleration and design can be implemented in a FPGA.

Researchers compare with the performance of all kinds of feature descriptors, including Harris method[7], Shi-Tomasi method [8], Speeded-Up Robust Features (SURF) [9], Scale Invariant Feature Transform (SIFT) [10], GLOH [11], PCA-SIFT [12], etc., in the affine transformation. The experiment results show that SIFT descriptors perform best. And become the most popular image stitching algorithm. Because of invariance on the scale, rotation and the change of illumination, and regardless of the outstanding performance of the SIFT feature descriptors on distinctiveness and robustness. However, the conventional of SIFT requires high computational complexity, especially in SIFT's Difference of Gaussian (DoG) process. Most experiments result shown that it is very difficult to perform on original PC by even SURF algorithm in real-time. It's a hot topic to solve this difficult problem. Sheng Zhong [13] proposed a real-time embedded architecture for SIFT algorithm, the proposed FPGA + DSP architecture design integrates the parallelism of FPGA and the flexibility of the high-performance DSP. Moreover, Vanderlei Bonato [14] put forward parallel hardware architecture for Scale and Rotation Invariant Feature Detection, and the method can be applied to the Simultaneous Localization and Mapping problem. Murad Qasaimeh [15] proposes parallel hardware architecture for real-time image classification based on FPGA. The parallel hardware architecture is able to detect up to 1270 SIFT features per frame. In this paper, we can make use of the powerful parallel processing characteristics of FPGA to achieve image feature detection in real-time based on SIFT algorithm.

The rest of paper is organized as follows: Section II describes a brief introduction of sift algorithm. Section III explains the parallel hardware implementation of sift algorithm. Section IV reviews experiment results and performance comparisons. Finally, the paper comes to a conclusion in Section V.

II. A BRIEF INTRODUCTION TO SIFT ALGORITHM

The real-time of feature detection system is mainly completed on FPGA, PC machines and special display monitor, shown as Figure 1. PC is primarily responsible for the procedures preparation, simulation, download and online commissioning. The main subject of FPGA consists of image acquisition, feature detection and I2C bus control.

The embedded system is mainly composed of three parts: video acquisition module, SIFT algorithm module and real-time display module.

- Video acquisition module: VITA2000 CMOS as the video input, in order to meet certain regulatory requirements, image format conversion is necessary.
- SIFT algorithm module: This module is mainly responsible for feature detection of input image, main

J.Q. Peng, Y.H. Li, W.G. Zhou, K. Fan is with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen Graduate School (e-mail: pjqxs@163.com).

C.Y. Lyu is with School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China. The Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong (e-mail: lylvcongyi@ 126.com).

Y.H. Liu is with The Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong.

978-1-4673-8959-4/16/$31.00 © 2016 IEEE

direction distribution and feature point descriptor calculation.

- Real-time display module: The module is primarily responsible for video real-time display.

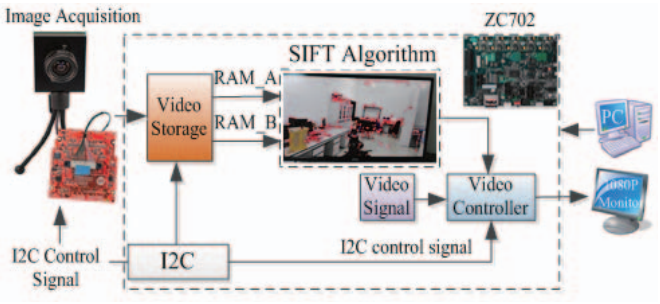

Figure 1. The overview of the embedded system

The most important task is SIFT algorithm, which can reflect the distribution of gray and gradient. Multi-scale analysis method is utilized to extract the feature points which have a very good stability, and can adapt to the changes and interference between the images. The SIFT algorithm implementation is shown in Figure 2.

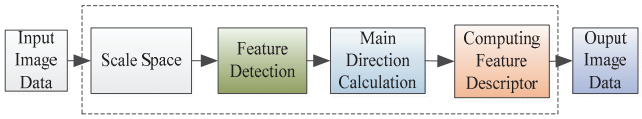

Figure 2. A block diagram of sift algorithm

Traditional SIFT algorithm can be decomposed into the five parts: DoG pyramid construction; establishing a DoG (Difference of Gaussian) scale space; in search of stable feature point; determining the gradient magnitude and orientation of the feature points; build feature descriptor.

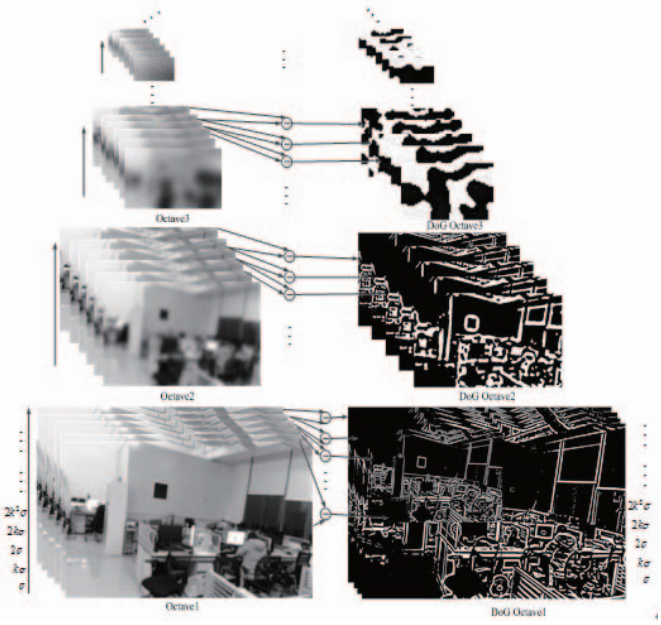

Figure 3. High-level diagram of the architecture for Gaussian pyramid generatio

From the parallel architecture for the overall system, we know that the most important task is build the DoG pyramid, the process of DoG pyramid construction is shown in Figure 3.

In this paper, there are nine octaves, each having been further divided into six Gaussian-filtered images and five DoG images.

1) Build a Gaussian Pyramid

Establish a Gaussian pyramid is to establish a DoG scale space. In SIFT algorithm, we can produce a variety of different scales Gauss image by Gaussian convolution of the original image. They can be defined as:

$$L(x,y,\sigma)=G(x,y,\sigma)\times I(x,y) \tag{1}$$

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

Where, $G(x,y,\sigma)$ is σ scale Gaussian Function, $I(x,y)$ is the input image, $L(x,y,\sigma)$ is σ scale space.

2) Establish a DoG Scale Space

Using the generated Gaussian pyramid, we can get a group of DoG images by subtracting its adjacent scale space image:

$$\begin{aligned} D(x,y,\sigma)&=G(x,y,k\sigma)\times I(x,y)-G(x,y,\sigma)\times I(x,y)\\ &=L(x,y,k\sigma)-L(x,y,\sigma)\end{aligned} \tag{3}$$

In DoG scale space, where the minimum point or maximum point act as a feature point. It is concluded that the local extreme value point coordinates and scale is the feature point coordinates and scale.

3) Stable Keypoint Detection

After the DoG scale space has been established, we can acquire the local maximum and minimum point in the DoG images by comparing a pixel to its neighbors at the adjacent scales. And we regard these local maximum and minimum points as candidate features.

In order to eliminating the low contrast points and strong edge response points, making the feature points are of very high robustness is very important. After eliminated the low contrast points, we should computed Hessian matrix $\boldsymbol{H}$, it defined as

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{D}_{xx} & \boldsymbol{D}_{xy}\\ \boldsymbol{D}_{yx} & \boldsymbol{D}_{yy}\end{bmatrix} \tag{4}$$

Here, $\boldsymbol{D}_{xy}$ is estimated by taking the difference of neighbor points, and the second order derivative in the x and y direction.

If we defined $\sigma=\lambda_{\max}/\lambda_{\min}$, then, we get

$$\frac{Tr(\boldsymbol{H})^2}{Det(\boldsymbol{H})}=\frac{(\lambda_{\min}+\lambda_{\max})^2}{\lambda_{\min}\lambda_{\max}}=\frac{(1+\sigma)^2}{\sigma} \tag{5}$$

Where, $Tr(\boldsymbol{H})$ means the trace of $\boldsymbol{H}$, and $Det(\boldsymbol{H})$ means the determinant of $\boldsymbol{H}$.

Therefore, so as to eliminate strong edge response points, we only meet the following requirements, that is to say:

$$\frac{Tr(H)^2}{Det(H)}\le\frac{(1+\sigma)^2}{\sigma} \tag{6}$$

4) Gradient Magnitude and Orientation Assignment

In order to make the feature descriptor has better stability, through computing pixel gradient direction for the main direction, the key point of a pixel gradient direction is one of the eight directions, gradient modulus value $m(x,y)$ and gradient direction $\theta(x,y)$ can be defined as:

$$m(x,y)=\sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2+\left(L(x,y+1)-L(x,y-1)\right)^2} \quad (7)$$

$$\theta(x,y)=a\tan 2\left(L(x+1,y)-L(x-1,y),L(x,y+1)-L(x,y-1)\right) \quad (8)$$

5) Build Feature Descriptor

The description of feature points you can refer to Figure 4. The left figure is the feature adjacent blocks of the image gradient, contains the information such as the amplitude, direction of pixels, while the right is the feature descriptor after calculate the gradient direction histogram, which consists of four seeds information point.

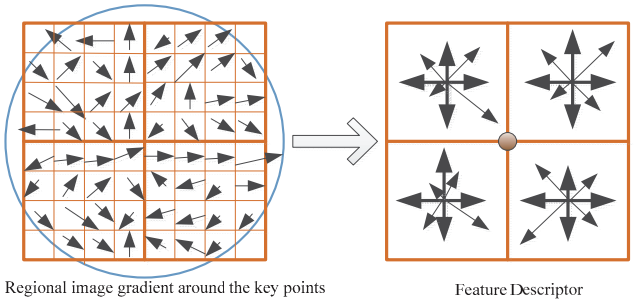

Figure 4. Schematic diagram of feature points description

III. A Parallel Hardware Architecture for SIFT Algorithm

The implementation process of Sift algorithm based on FPGA is shown in Figure 5.

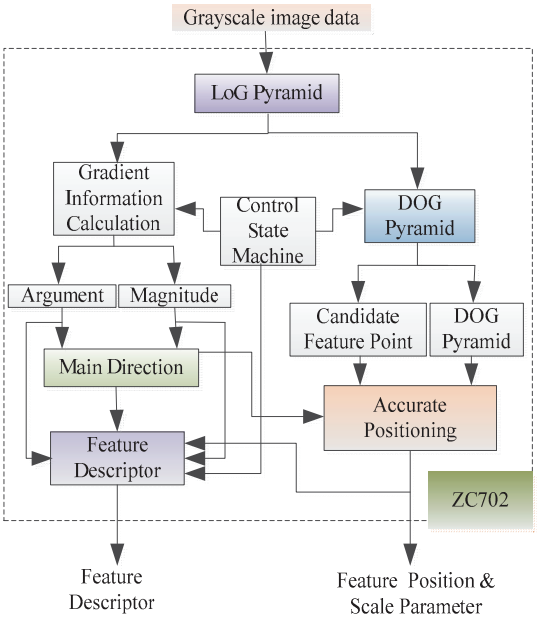

Figure 5. Schematic diagram of sift algorithm based on fpga

1) Parallel Structure Design for Gaussian Filter

Gaussian weighted weight is a Gaussian function spatial discretization in two-dimensional, and the two-dimensional Gaussian image filter completed by the convolution of two-dimensional discrete weight and image. Since the filtering effect of the two-dimensional Gaussian convolution filter can be approximated by a one-dimensional Gaussian filter in two different directions, in order to save hardware resources, here, we use one-dimensional filtering on the image x, y direction.

This paper chooses 7×7 filtering window, the specific implementation is the line Gaussian filter in the x direction, and the column Gaussian filter in the y direction. Gaussian parallel design flow is shown in Figure 6. Here, D0, D1, D2, D3, D4, D5, D6 are the seven sets of D flip-flops, each group comprises 12 D flip-flop. G0、G1、G2、G3、G4、G5 and G6 are the coefficients for one dimensional Gaussian filter.

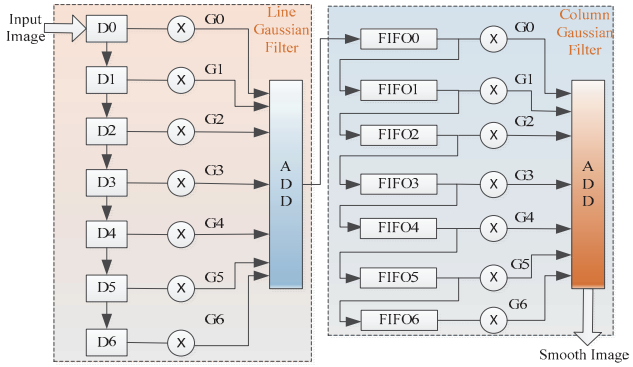

Figure 6. Parallel processing flowchart for two-dimensional Gaussian filter

2) Parallel Structure Design for DoG Pyramid

As Figure 7 puts, after the image data stream flow into DoG Pyramid, a serial continuous Gaussian filtering, due to the relative timing of the time delay, the adjacent Gaussian filtering images cannot be directly subtracted. In order to achieve the above mentioned delay, here, we using five rows FIFO and the depth of each FIFO are three lines of image size.

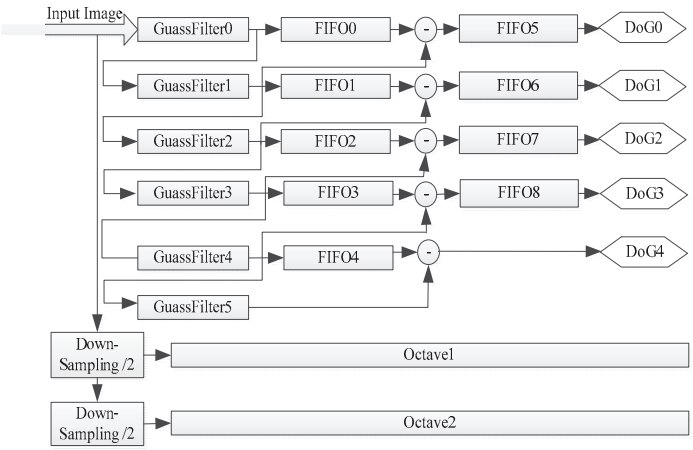

Figure 7. Parallel design flow for DoG Pyramid

3) Implementation of Stable Keypoint Detection Module

In this section, the feature point extraction consists of two parts: one is the parallel structure design for obtaining extreme points; the second is the structure design for removed feature points modular. As the second chapter puts, the extreme point mainly is carried in three adjacent DoG pyramid image. The detail parallel implementation is shown in Figure 8. For each DoG pyramid group is consist of five layers DoG diagram, there are three layers produced the extreme points, the extreme points correspond to D15, D25 and D35, respectively. Weather D15 is the extreme point or not, you can make comparison with 26 points around D15. Similarly, for each set of images can produce three-layer extreme points, three groups DoG diagram are generating nine-layer extreme points. A brief introduction of feature extraction is shown in Figure 9.

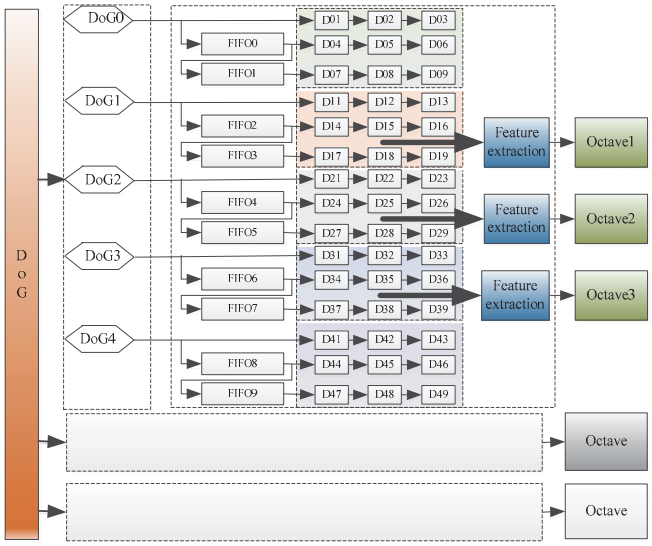

Figure 8. Parallel structure block diagram design for Feature positioning

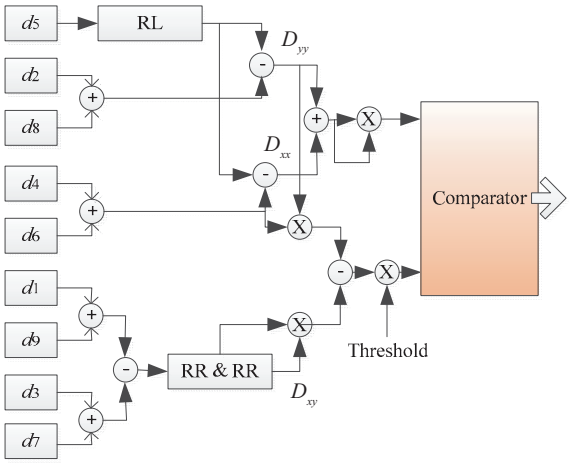

Figure 9. The flowchart of feature extraction module

4) Gradient Magnitude and Argument Computation Module

In order to solve the problem of image pixel gradient, here, we use Cordic algorithm. Cordic algorithm is suitable for sine, cosine, arctan function. Since Cordic algorithm only use shift and addition operations, which greatly facilitates the hardware implementation of these complex nonlinear functions, and widely used in the hardware system. In Cordic rotation process, FPGA implementation uses 11 pipeline, fully meets the requirements. The last pipelined output is the next pipeline input. The structure of 11 pipelines is shown in Figure 10.

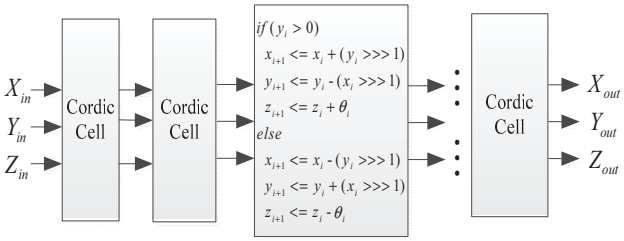

Figure 10. The flowchart of Cordic Line array

5) Feature Descriptor Calculation Module

As the last section puts, each key point has three corresponding information, namely: the position coordinates, scale parameter and the main direction. You need to use the information to set up the corresponding feature descriptor, so that the invariant characteristic description of key points, here, we defined 128-dimensional descriptor vector in sift algorithm, the vector contains feature neighborhood information, has the high uniqueness, and lay a foundation for feature matching in SIFT algorithm. Architecture design for feature descriptor calculation module is shown in Figure 11.

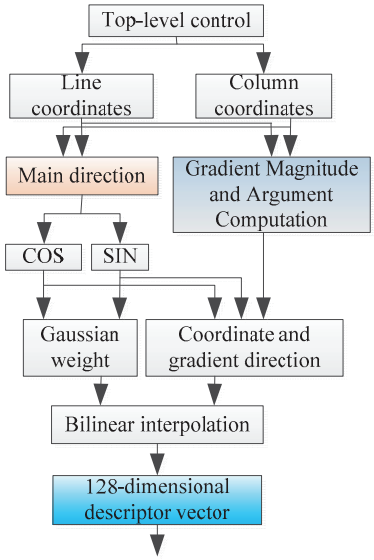

Figure 11. The flowchart of feature descriptor calculation module

IV. Test and Verification

A. Experimental Results

In this paper, the camera resolution is 1920*1080, the SIFT algorithm is implemented in Xilinx ZC702 hardware platform, using VIVADO 2014.4 software to write the Verilog language; our proposed hardware is able to detect any number of features at 30 frames/second. The SIFT algorithm is completed on PL; our development platform is presented in Figure 12. Based on the parallel hardware algorithm, the

proposed hardware performs at 30 fps keypoint extraction, shown in Figure 13; moreover, we have verified that it is possible to perform keypoint matching up to 1102 keypoints on the FPGA.

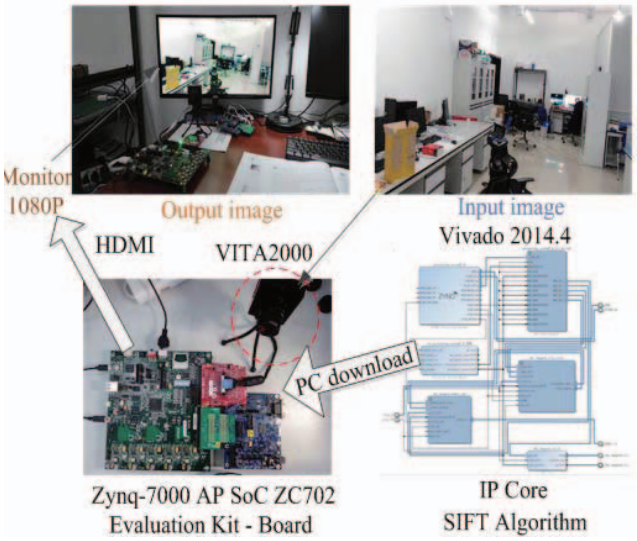

Figure 12. The implementation of our proposed architecture: the FPGA coupled with ARM platform

Figure 13. The demonstration of feature point detection in four different directions

B. *Performance Comparison*

In order to test the real-time performance of SIFT algorithm; in this paper, we estimated the computation time of the ZC702 platform and PC platforms (The development environment on software is Visual Studio C++ 2010 and Matlab 2013, CPU is Intel Core i7 CPU-4790 3.60 GHz). The performance comparison results showed in TABLE I.

TABLE I. THE COMPUTATION TIME OF FPGA AND PC PLATFORMS

Name	FPGA and PC Comparison		
	ZC702	**VS2010**	**Matlab2013**
time	0.05sec	26.89sec	87.37sec
Image size	1920*1080	1920*1080	1920*1080

keypoints	1102	1447	1523

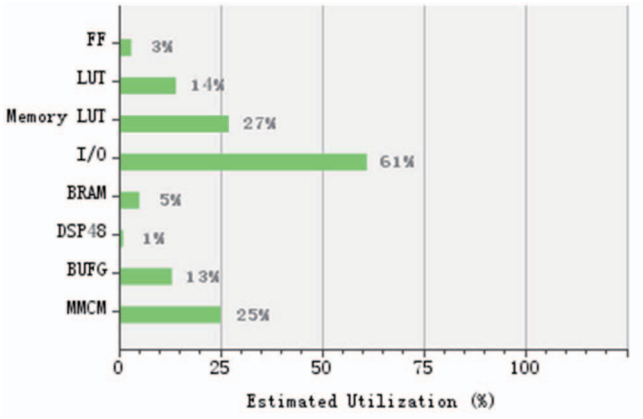

Figure 14. The estimated utilization of sift algorithm

As Figure 14 puts, the algorithm running on FPGA resource consumption rate is low; high real-time performance and the design accuracy meet the matching requirements. Experiment results mainly are reflected in two aspects.

In this respect, the accuracy of feature points and the sub-invariance verified the accuracy of the SIFT algorithm.

On the other hand, compared with the frequency running the SIFT algorithm on Intel Core i7 CPU-4790 3.60 GHz, the computation time running on ZC702 hardware platform increased by nearly two orders of magnitude.

V. CONCLUSION

On the basis of traditional SIFT algorithm, the main disadvantages of it is its high computational cost, therefore, this paper propose an efficient hardware architecture based on FPGAs for SIFT algorithm. The main contribution of the hardware architecture proposed in this paper is as follow: Comparing with SIFT algorithm running on PC, the parallel algorithm not only makes feature detection more accurate, but also makes feature extraction faster. Moreover, the proposed parallel hardware architecture is more satisfactory in time and accuracy, which meets the real-time requirements in our future computer vision applications.

The major future work may be including: (1) accelerate the process of SIFT feature description; (2) improve the matching accuracy.

ACKNOWLEDGMENT

The work supported by Shenzhen Peacock Plan Team grant KQTD20140630150243062, Shenzhen Fundamental Research grant JCYJ20140417172417120 and JCYJ20140417172417145, Shenzhen Key Laboratory grant ZDSYS20140508161825065.

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.

[2] L. Chang, L. E. Sucar. “FPGA-based detection of SIFT interest keypoints,” *Machine Vision and Applications*, vol.24, pp.371-392, 2013.
[3] J. Zhu andM.w. Ren, “Image Mosaic Method Based on SIFT Features of Line Segment,” *Computational and Mathematical Methods in Medicine*, vol.1, no.1,pp.1-11,2014.
[4] N. Chiba, H. Kano, “Feature-Based Image Mosaicking,” *Systems and Computers*, vol.31, no.7, pp.1581-1589,2000.
[5] 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.
[6] J. Herbert, S. Wilson, “FPGA implementation of a high-speed, real-time, windowed standard deviation filter,” *Electronics Letters*, vol.34, pp. 22–23,2016.
[7] J. Zhu, M.W. Ren, “Fast matching algorithm based on corner detection,” *Journal of Nanjing University of Science and Technology*, vol. 35, no. 6, pp. 755–758, 2011.
[8] J. Shi , C. Tomasi, “Good features to track,” in *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.
[9] H. Bay, T. Tuytelaars, and L.V. Gool, “SURF: Speeded up robust features,” *ECCV*, pp.404–417, 2006.
[10] G. Michael, G. Helmut, and B. Horst, “Fast approximated SIFT,” *Proc. ACCV*, pp.918–927, 2006.
[11] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell*., vol.10, no.27, pp.1615–1630, 2005.
[12] Y. Ke, R. Sukthankar, “A more distinctive representation for local image descriptors,” *Proc. Computer Vision and Pattern Recognition*, pp.506–513, 2004.
[13] S. Zhong, J.h. Wang, “A real-time embedded architecture for SIFT,” *Journal of Systems Architecture*, vol.59, pp. 16–29, 2013.
[14] V. Bonato, E. Marques, “A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection,” *IEEE Transactions On Circuits And systems For Video Technology*, vol. 18, no. 12, December 2008.
[15] M. Qasaimeh, A. Sagahyroon, “FPGA-Based Parallel Hardware Architecture for Real-Time Image Classification,” *IEEE Transactions On Computational Imaging*, vol. 1, no. 1, pp, 56-70, March 2015.