Design of a High Speed 360-degree Panoramic Video Acquisition System Based on FPGA and USB 3.0

Congyi Lyu, Jianqing Peng, Weiguo Zhou, ShanshanYang and Yunhui Liu, *Fellow, IEEE*

***Abstract*—**This paper presents a FPGA-based hardware architecture for capturing real-time panoramic images and the parallel implementation of an image stitching algorithm based on the scale-invariant feature transform (SIFT) features. First, the system transfers the images captured by five CMOS sensors to the designed FPGA board through the FMC adapter plate. Then, the system employs the FPGA-based parallel computing to extract the SIFT features and detect the best matching region and points for image registration. Finally, in order to obtain the panoramic image with smooth changes at the junctions, images of the adjacent images are fused. Using the designed FPGA platform makes it possible to extract the SIFT and stitch the images in real-time, which is not realized in most panoramic imaging systems. The panoramic system, developed using the Xilinx Zynq®-7000 all programmable SOC, can capture video of 5*640*480 images at the speed of 60 fps. To enable high-speed image transfer from the imaging system to the host computer, the USB3.0 interface with the transmission speed of 359MB/s has been developed. The real-time efficiency of the parallel system will be demonstrated by the simulations and experiments.

***Index Terms*—Panoramic images; FPGA-based parallel computing; Image mosaicking; Embedded systems.**

I. INTRODUCTION

PANORAMIC STITCHING, the process of achieving a wider field-of-view of a scene from a set of images, has been widely used in various applications, including mapping, medical imaging, motion tracking, scene reproduction for computer vision etc. Panoramic stitching can reduce the redundant information in multiple sets of images, increase the image storage capacity, and enable more effective views of a real world. [1][2][3][4][5][6].

Ghosh and Debabrata[7] conducted a comprehensive survey on image mosaicing techniques and pointed out the disadvantages and advantages of existing methods. The basic idea in image mosaicing is to detect common features in two adjacent images and then to stitch them together based on the common features. The features used for this purposes include the Harris corner [8], the good features developed in [9], the SIFT [10], the speeded-up robust features [11], the GLOH [12], etc. Zhu and Ren [13] introduced an image mosaicing algorithm based on the SIFT features on line segments. The proposed method showed strong robustness to scaling, rotation, resolution and illumination. Xie et al. [14] proposed an improved image mosaic method which combines the wavelet transform and the compressed sensing algorithm and demonstrated the improvement of the stitching results by experiments. However, the major concern is that all the mosaicing techniques running on PC platforms are time consuming and hence cannot generate real-time performance. There are two ways to solve the problem: to improve the efficiency of the algorithms or to use hardware to accelerate the computation. Despite of tremendous efforts on the development of real-time algorithms, the progress is limited.

The research effort on hardware optimization is mainly made to parallel implementation of the algorithms using GPU, DSP, or FPGA[15]. Using GPU leads to a big system and large consumption. For its low power consumption and high flexibility, FPGA or FPGA plus DSP is popularly used in parallel computation [16]. Qasaimeh and Sagahyroon [17] developed a parallel hardware architecture for real-time image classification based on FPGA. To achieve a real-time performance, the proposed architecture accelerates their execution by exploiting different forms of parallelism in these algorithms. Wang and Zhong [18] presented an embedded system for feature detection and matching based on FPGA, The developed system produced real-time performance in detecting features of images captured by a 720p-camera with 60 fps. They further developed, in [19], a real-time embedded architecture for implementing the SIFT algorithm. This hardware design integrates the parallelism of FPGA and the flexibility of the high-performance DSP. For comparison, Bonato and Marques [20] built another parallel hardware architecture for scale and rotation invariant feature detection.

However, the works mentioned above are for real-time processing of images from a single camera. To generate

C. Y. Lyu is with Intelligent Robotics Institute, Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China (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 (e-mail: yhliu@mae.cuhk.edu.hk)

J. Q. Peng, W. G. Zhou and S. S. Yang are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen Graduate School (e-mail: pjqxs@163.com, zhouweiguo1991@gmail.com, ZWGUO2008@126.com).

panoramic images, it is necessary to process images from multiple cameras. In this paper, we present a parallel system for real-time capturing, processing, and transmission of images from five cameras. Since the SIFT descriptors [21] have its outstanding performance of robustness and distinctiveness, we employ SIFT features as the key points in the panoramic stitching algorithm. In order to realize real-time image transmission between the system and a host PC, the USB 3.0 interface is used in the parallel system.

The rest of paper is organized as follows: Section II presents an overview of the system and Section III gives the hardware implementation of the SIFT-based panoramic stitching algorithm. We will demonstrate the experimental results in Section IV and conclude the work in Section V.

II. Overview of the Panoramic Mosaicking System

As shown in Fig. 1, the proposed panoramic video acquisition system is composed of a PC, a ZYNQ-7000 board, USB 3.0 interface and a 1080-P monitor. The simulation of the panoramic image stitching algorithm is conducted on the PC using the software named Vivado2014.4. After synthesis and implementation of the parallel algorithm, the software will generate the bitstream file, which is exported to the to Vivado SDK software. By adding the register driver for Dual Arm Cortex A9MP hard cores, downloading the software to and then running on the designed ZYNQ-7000 board, the panoramic image will transmitted to PC host from FPGA board via the USB 3.0 interface.

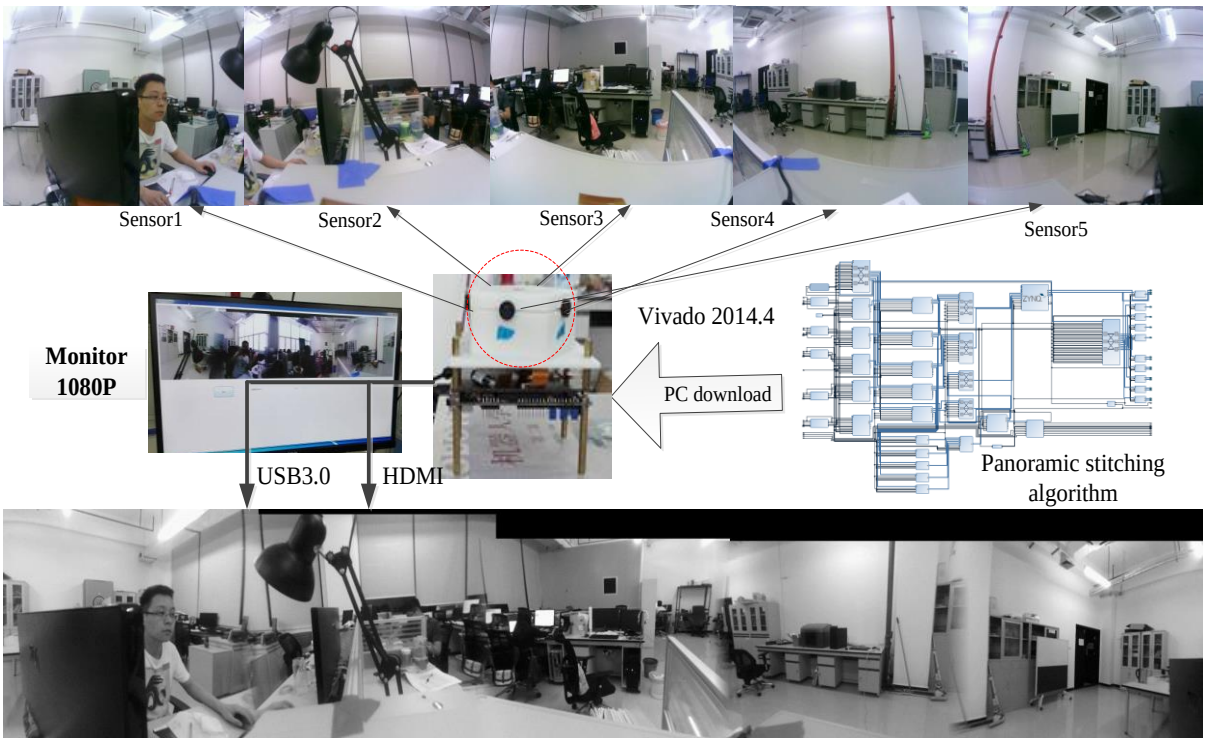


Fig. 1. Overview of panoramic stitching system.

The panoramic video image acquisition system can be classified into six functional modules: image acquisition module, feature extraction module, feature matching module, image fusion module, USB 3.0 interface and HDMI displaying module.

Image acquisition module: Obtain stable images from the CMOS sensor OV5640, and the inner registers can be reconfigured by the I2C port.

Feature extraction module: Extract the SFIT features of the images using the designed FPGA-based parallel computing platform.

Feature matching module: Calculate the similarity of two feature descriptors and establish the correspondence by the nearest neighbor ratio method.

Image fusion module: Fuse images of two adjacent images in order to obtain smooth transition effects between the two images whose brightness and orientations are different.

USB 3.0 interface: This module enables real-time tranfer of the big image data from the system to a host computer via the USB 3.0 interface.

HDMI displaying module: Display the real-time panoramic stitching results on a 1080-P monitor.

The detailed flow of the panoramic image stitching process is shown in Fig. 2. First, after the ZYNQ receives the image data (5 @640*480), we store the data into DDR3 through five VDMA. Second the SIFT features of the images are detected by parallel computation. Third, we compare the key points and choose the best matching points by the nearest neighbor ratio method. Finally, we fuse the five images to obtain the smooth panoramic image, which is displayed on a 1080P monitor with the HDMI interface. The panoramic image is transmitted to the PC host in real-time through the USB 3.0 interface.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2628240, IEEE Sensors Journal

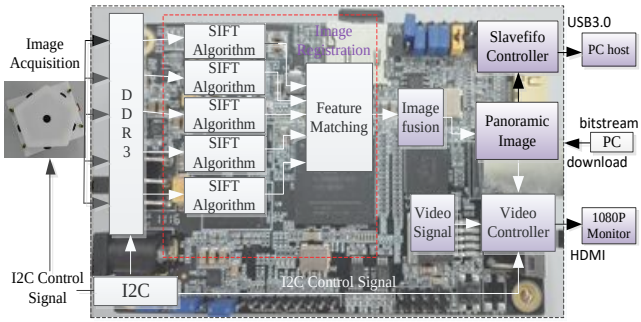


Fig. 2. The flow of the image mosaicking system.

III. Hardware Implementation of the Panoramic Stitching Algorithm

To stitch two images, the most important task is image registration, which is to register the same scene or points in the two images. We use the SIFT features as the points to be registered. The key step of image registration is shown in Fig.3. Using SIFT features can help improve the robustness to the image noises. The multi-sale analysis method is used to detect stable features so as to register the images more stably and accurately.

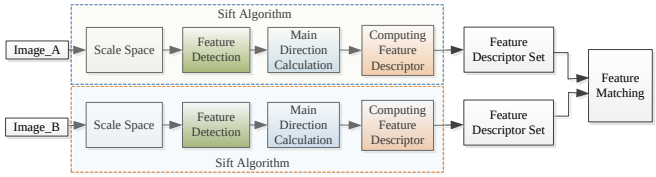


Fig. 3. A block diagram of the image registration process.

We know that SIFT algorithm is proved to be the most robust feature, which can reflect the distribution of the gray and gradient. Multi-scale analysis method is utilized to extract the feature points which have good stability, and to adapt to the changes and interference between the images.

A. Architecture Design of the SIFT Algorithm

The implementation process of SIFT algorithm based on FPGA is shown in Fig. 4.

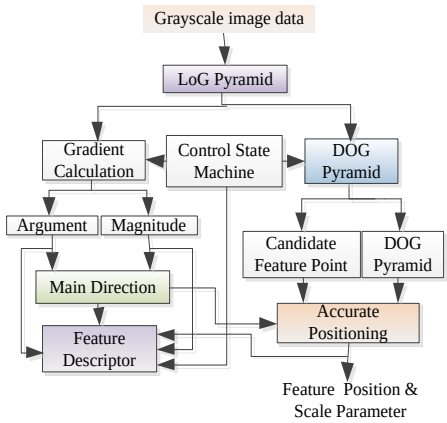


Fig. 4. The schematic diagram of the SIFT algorithm based on FPGA.

The SIFT algorithm consists of five parts: building a Gaussian pyramid; establishing a DoG Gaussian scale space; searching for key points; calculating the gradient direction; building the feature descriptors.

1) *Parallel Structure Design for DoG Pyramid*

By carrying out the Gaussian convolution of the input image, we can obtain a variety of Gauss images with different scales. This step can be determined by:

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

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

$G(x,y,\sigma)$ is the σ -scale Gaussian function; $I(x,y)$ is the input image; and $L(x,y,\sigma)$ is the σ -scale Gaussian image.

The effect of the two-dimensional Gaussian convolution filter can be approximately conducted by two one-dimensional Gaussian filters in two different directions. Considering the limited hardware resources, we choose a 7×7 filtering window in which one-dimensional row Gaussian filtering and one-dimensional column Gaussian filtering are conducted in the x and y directions, respectively. The logic schematic of the parallel Gaussian filtering is shown in Fig. 5. We need to point out that $D_i(i = 0\sim6)$ are the seven sets of D flip-flops, and $G_i(i = 0\sim6)$ are the coefficients of one dimensional Gaussian filter.

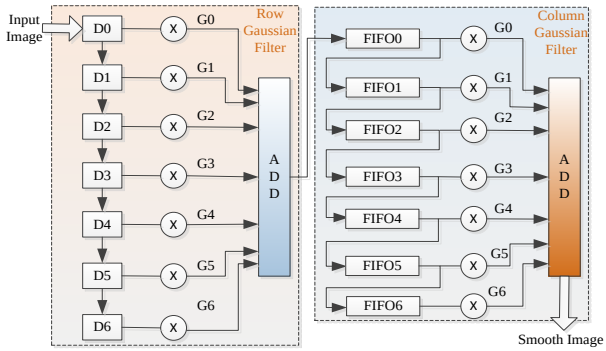


Fig. 5. The parallel design for a two-dimensional Gaussian filter.

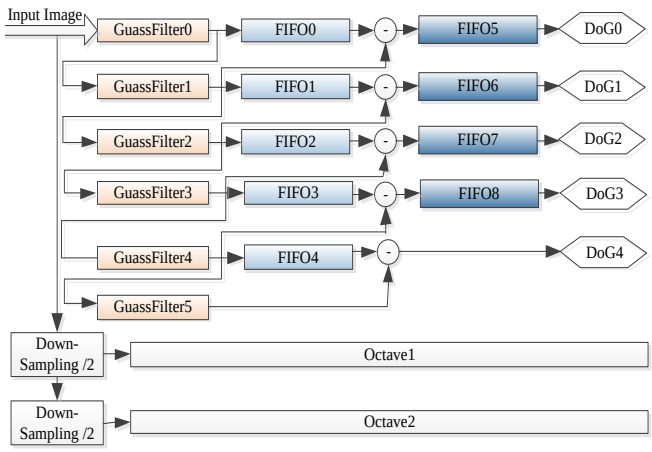


Fig. 6. Parallel design of the DoG Pyramid.

Due to the differences in timing delay, the different Gaussian images cannot arrive the next processing step at the same time. Eight FIFO devices were used to provide an additional data transferring delay which is shown in Fig.6. By this way, the different Gaussian images can be processed at

the same time.

2) Establishing a DoG Scale Space

After the Gaussian pyramid was generated, the DoG (Differential of Gaussian) images were conducted by subtracting the Gaussian images in the adjacent Gaussian scale space. This calculation can be defined as:

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

Here, we take the image data grabbed by the first CMOS sensor as an example. Its DoG Gaussian scale space is shown in Fig. 7.

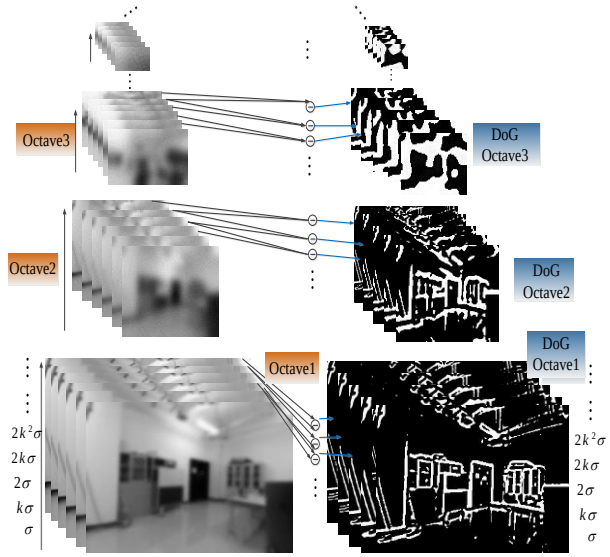


Fig. 7. The process of generating a DoG Gaussian scale space.

3) Implementation of Stable Key Points Detection Module

After establishing the DoG Gaussian scale space, we can extract the local maximum and minimum points in the DoG images, which are considered as the candidate feature points.

In order to extract more robust feature points, the low contrast points and the strong edge response points need to be removed. We define the following Hessian matrix $\boldsymbol{H}$:

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

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

If we defined $\sigma = \lambda_{\max} / \lambda_{\min}$, to eliminate the strong edge response points, we should propose the following criterion:

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

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

The flowchart of the feature extraction module is shown in Fig. 8. This module is designed to remove unstable key points.

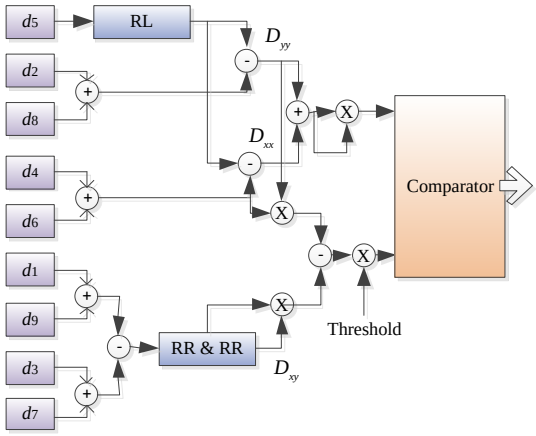


Fig. 8. The flowchart of the feature extraction module.

The feature point extraction is composed of two parts: to obtain the extreme points and to remove the disturbed feature points. The extreme points are detected using three adjacent DoG pyramid images. As Fig. 9 shows, each DoG pyramid group consists of five layered DoG images, within which only three produce the extreme points, D15, D25 and D35, respectively. To determine whether D15 is the extreme point or not, we need compare with the neighboring 26 points around D15.

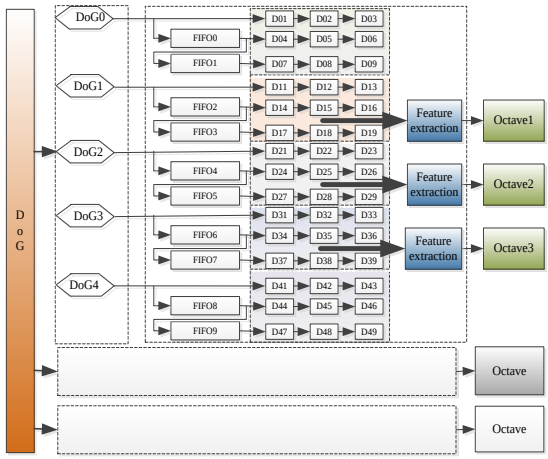


Fig. 9. The parallel structure for the stable key points detection.

4) Gradient Magnitude and Orientation Assignment Calculation

In order to extract the stable feature descriptor, we need to compute the main direction. The gradient magnitude $m(x,y)$ and the gradient direction $\theta(x,y)$ can be defined as:

$$m(x,y) = \sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2 + \left(L(x,y+1)-L(x,y-1)\right)^2} \tag{6}$$

$$\theta(x,y) = a\tan 2\left(L(x+1,y)-L(x-1,y), L(x,y+1)-L(x,y-1)\right) \tag{7}$$

Since the CORDIC algorithm is suitable for sine, cosine and arctan functions, so we can use it to solve the main direction. The CORDIC algorithm only uses shift and addition operations, which greatly facilitate the hardware implementation of these complex nonlinear functions. The

structure of the CORDIC line array is shown in Fig. 10.

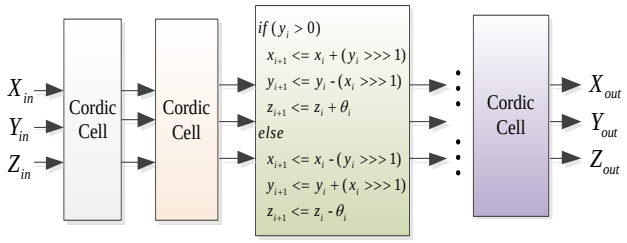


Fig. 10. The flowchart of the CORDIC line array.

5) Building Feature Descriptors

As Fig. 11 shows, the left figure is the feature gradient block, containing the amplitude and pixels' directions. The right is the feature descriptors containing the gradient directions.

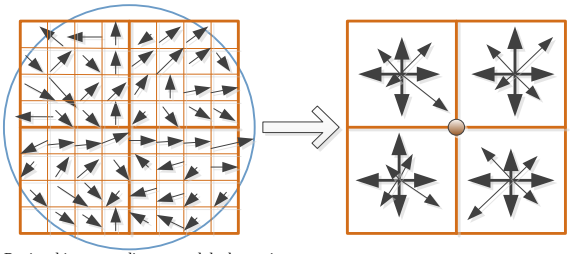


Fig. 11. Schematic diagram of the feature points description.

As mentioned at previous sections, each key point has three types of corresponding information: the position coordinates, the scale parameter and the main direction. The feature descriptor was defined as a 128-dimensional vector, which contains the feature of the neighborhood pixels. The architecture design of the feature descriptor calculation module is shown in Fig. 12.

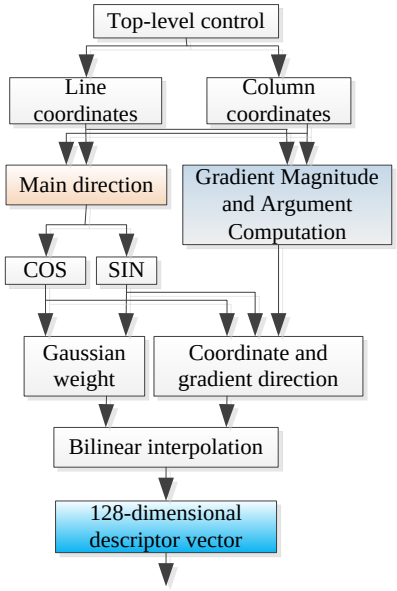


Fig. 12. The flowchart of feature descriptor calculation module.

The demonstration of the feature point detection module on the images captured by five imaging sensors is shown in Fig. 13.

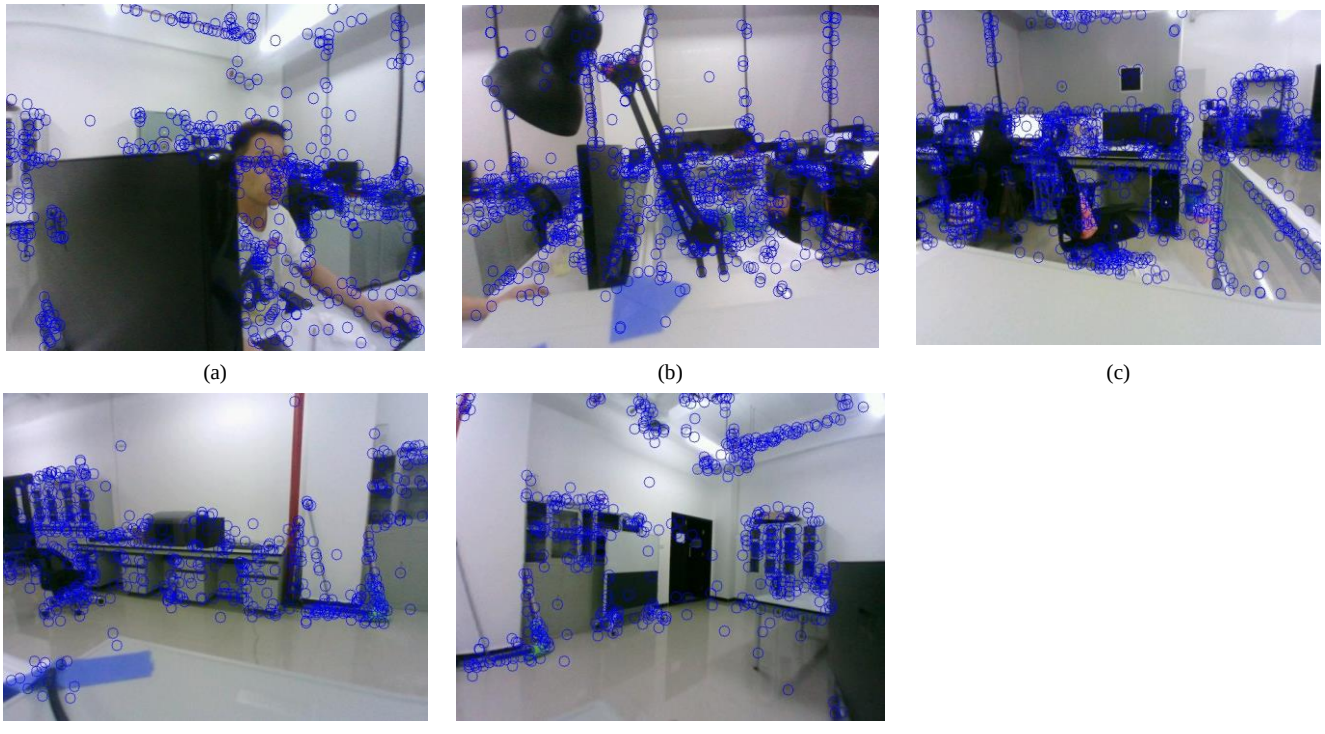


Fig. 13. Feature point detection for sensor1~ sensor5: (a) Feature point detection for sensor1; (b) Feature point detection for sensor2; (c) Feature point detection for sensor3; (d) Feature point detection for sensor4; (e) Feature point detection for sensor5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2628240, IEEE Sensors Journal

B. Parallel Design for Feature Matching

After obtaining feature vectors in image_A and image_B, similarity of the two images was determined by using the nearest neighbor distance method. Given the feature descriptor D_m and D_n, the Euclidean distance can be calculated as:

$$d = \sqrt{\sum_{i=0}^{127} \left(D_m(i) - D_n(i) \right)^2} \quad (8)$$

The matching algorithm consumes much resource, so the matching operations can be implemented on the large scale FPGA device. The architecture of the feature matching is shown in Fig.14, and the matching results are shown in Fig. 15.

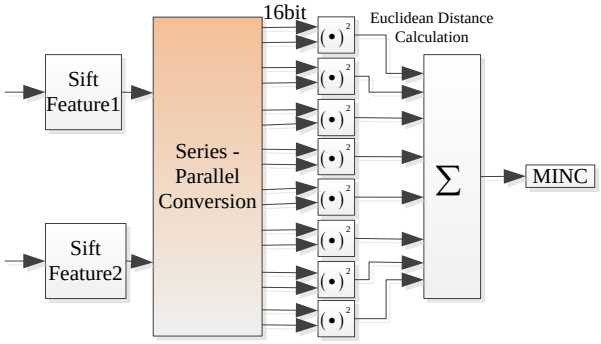
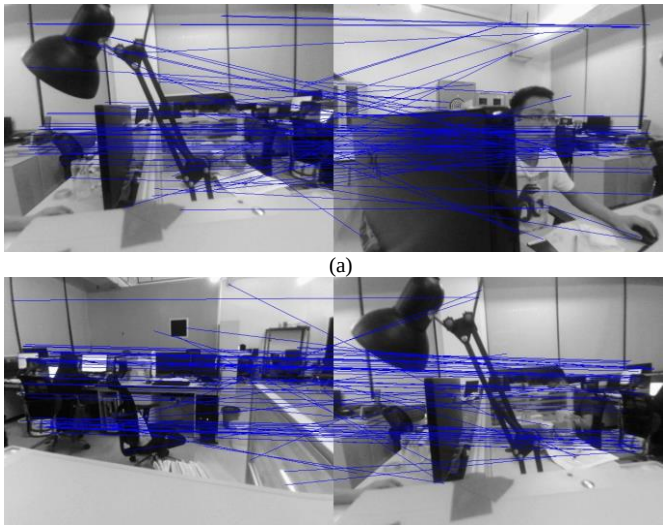


Fig. 14. The flowchart of feature matching module.



(a)

(b)

Fig. 15. Results of the matching: pairs (a) matching results of sensor1 and sensor2; pairs (b) matching results of sensor2 and sensor3.

C. Parallel Design for Image Fusion

In order to achieve a smooth mosaicing, we use the weighted smoothing algorithm to eliminate the echo problem. The idea of the smoothing algorithm is shown in Fig. 16.

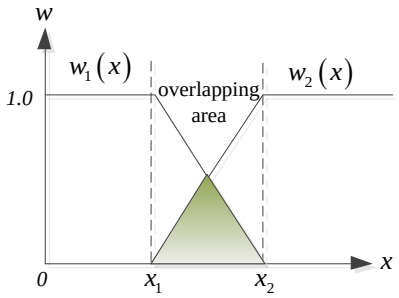


Fig. 16. Weighted smooth schematic diagram.

The overlapping area between the adjacent images I_1 and I_2 is in the interval $[x_1, x_2]$, and $w_1(x)$, $w_2(x)$ are the weighting factors which were defined as:

$$w_2(x) = 1 - w_1(x) = 1 - i/w \quad (9)$$

In equation(9), w is the length of the overlapping area, $0 \le i \le w$, the pixel value of the smoothed image I at position (x, y) can be calculated as:

$$I(x,y) = I_1(x,y) \times w_1(x,y) + I_2(x,y) \times w_2(x,y) \quad (10)$$

Some image fusion results are shown in Fig. 17 and Fig. 18.

Fig. 17. Results of image fusion by sensor1 and sensor2

Fig. 18. Results of image fusion by sensor1~ sensor3.

D. Implementation of High-Speed Transmission Module

We further designed the USB 3.0 interface for the panoramic camera to enable ultra-high speed image data transmission.

The USB 3.0 data transmission module works as follow: panoramic video passing through the GPIF circuit board to the FX3 device, which is configured as the slave FIFO mode; The received data were displayed on the host PC via Microsoft VS2010 software. The connection between ZYNQ-7000 and Cypress FX3 USB3.0 device is shown in Fig. 19.

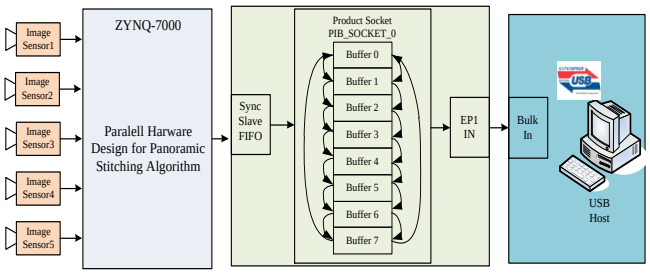


Fig. 19. Setup for FX3 transfer mode.

In order to display the image data at full resolution, additional work is necessary. The traditional method is adding serials of label, such as "5A5A5A5A". The efficiency of this method is low, and it was not suit able for the high-speed data transmission system. This paper solves this problem perfectly by using ZLP (zero length package) method. This method includes two steps: sending data stream and ZLP at the end of data stream.

To make sure the integrity of the frame, after sending the image data, the ZLP signal was transferred to notify the FX3 device the end of frame. The state machine and the control signals of the FX3 device were rested to the start time of the next frame.

The state machine for the FX3 Stream-IN mode is shown in Fig. 20.

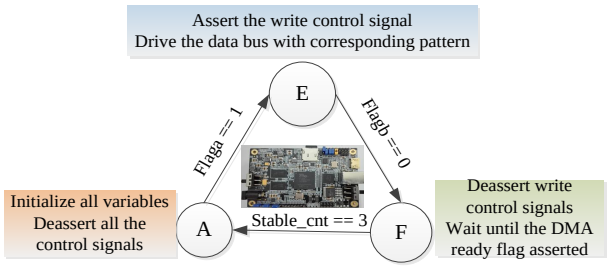


Fig. 20. FPGA State Machine for Stream IN mode.

In this mode, the status of the slave FIFO control signals in stage A is shown as follows:

$$PKTEND\# = 1;\ SLOE\# = 1;\ SLRD\# = 1;$$
$$SLCS\# = 0;\ SLWR\# = 1;\ A[1:0] = 0. \quad (11)$$

When Flaga = 1 the state machine of FX3 will enter into state E. The status of the data writing signal in state E is shown as follows:

$$PKTEND\# = 1;\ SLOE\# = 1;\ SLRD\# = 1;$$
$$SLCS\# = 0;\ SLWR\# = 0;\ A[1:0] = 0 \quad (12)$$

After 1 clock cycle of Flagb assertion, the sate machine of FX3 changes to state F. All the data writing signal are rested. Then the sate machine of FX3 changes to the ZLP mode, and the status is shown in Fig. 21.

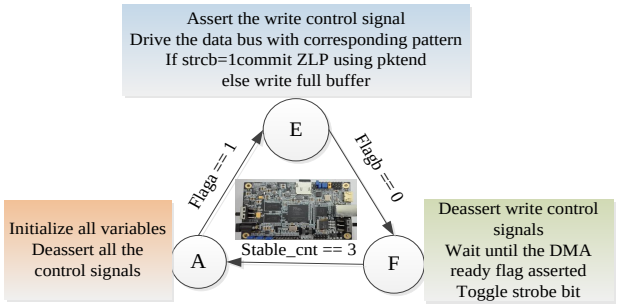


Fig. 21. FPGA State Machine for ZLP mode.

In state A, FX3 will initialize all the registers and signals used in the state machine. The status of the Slave FIFO control signals is shown as follow:

$$PKTEND\# = 1;\ SLOE\# = 1;\ SLRD\# = 1;$$
$$SLCS\# = 0;\ SLWR\# = 1;\ A[1:0] = 0 \quad (13)$$

When Flaga = 1 the state machine will enter into state E. In this state, if signal strob = 1 this package is considered as the ZLP package.

After 1 clock cycle of Flagb assertion, the sate machine of the FX3 changes to state F. All the control signals are rested. Then, the FX3 will enter into the Stream-IN mode.

IV. Experiments and Analysis

A. Experiment Results

The algorithm was implemented on a Xilinx ZYNQ-7000 FPGA board. There are four clock sources on this board: the pixel clock of the OV5640 image sensor is 50MHz; the system clock is 100MHz; the DDR3 read/write clock is 400MHz and the HDMI display module clock is 74.25MHz. The structures of the FPGA board and panoramic camera are shown in Fig. 22 and Fig. 23, respectively.

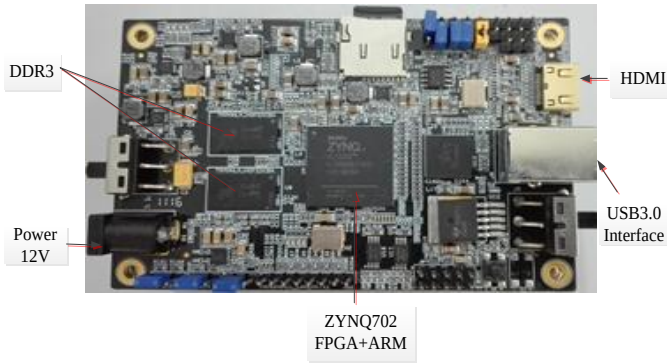


Fig. 22. ZYNQ-7000 serials development board.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2016.2628240, IEEE Sensors Journal

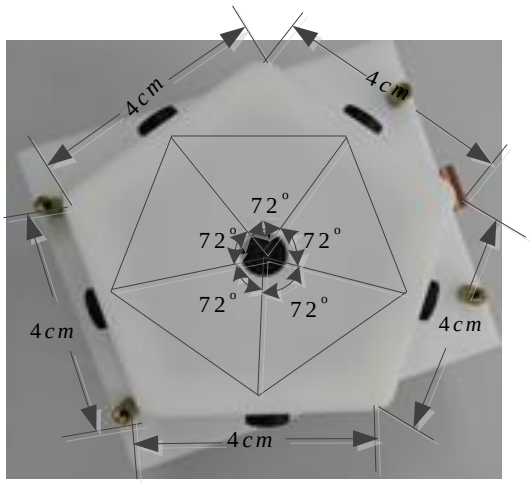


Fig. 23. The cameras' locations (side length = 4cm, height=6cm).

The panoramic camera system consists of three parts: a ZYNQ-7000 board, a panoramic camera and a HDMI display module. Fig. 24 shows a photo of the panoramic camera.

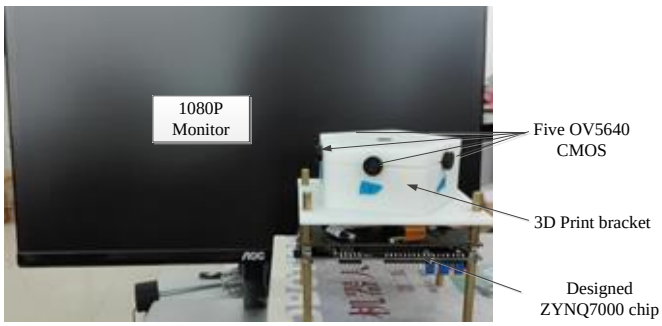


Fig. 24. Structure of panoramic camera system

The panoramic image mosaicking system is running on the SoC (System on Chip); the panoramic stitching algorithm is implemented on FPGA; and the registers are configured on an ARM core. The system uses five image sensors whose resolution is 640(H) × 480(V).

By using the proposed image stitching algorithm running on the ZYNQ-7000 board, we can obtain the panoramic image, which is shown in Fig. 25. The resolution of the panoramic image is 2745 (H) × 480(V); the frame rate can reach 33frame/sec, which meets the real-time requirements of the panorama camera display.

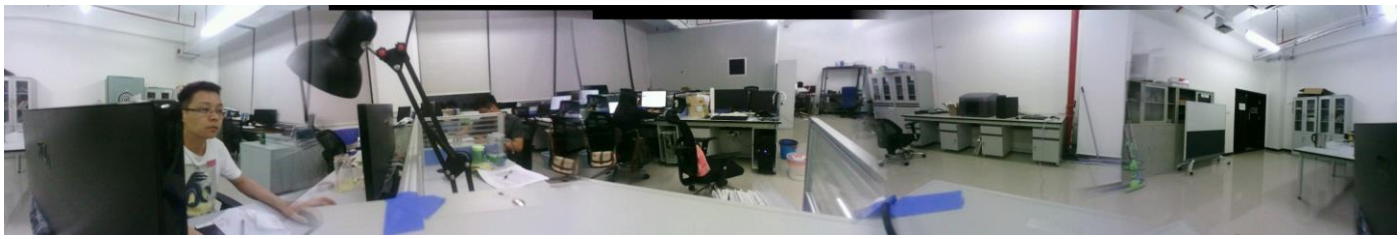


Fig. 25. 360-Degree panorama stitching results by the proposed panoramic system.

Using the USB3.0 interface, the transmission speed can reach 359MB/s, shown in Fig. 26, which is sufficiently fast for transmitting the panoramic images in real-time.

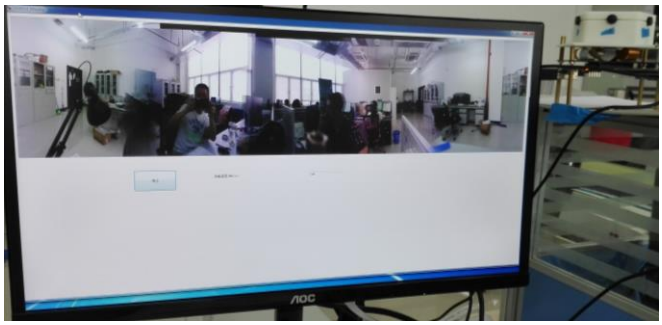


Fig. 26. The high-speed 360-degree panoramic video acquisition system based on FPGA and USB3.0.

B. Performance Analysis

In order to verify the real-time performance of the panoramic stitching system based on SIFT algorithm, we compared the computation time between the proposed ZYNQ7000 based platform and the PC based platform (Intel Core i7 CPU, 8G RAM). To ensure accuracy of the comparison, the experiment lasted 30 minutes the experimental results are shown in TABLE I.

TABLE I.

THE COMPUTATION TIME OF ZYNQ-7000 AND PC

Catagory	Image size	The average computation time		
		FPGA	*VS*2010	*Matlab*2013
Sift	640(H)*480(V)	4.08ms	3.75s	3.23s
Panoramic stitching	5×640(H)*480(V)	0.03s	1.21min	1.52min

As shown in Fig.27, the hardware resource utilization of the panoramic stitching algorithm is acceptable for the middle scale FPGA device. The input voltage and arrival current of the board are 12V and 3A, respectively. As shown in Fig. 28, the power consumption of the ZYNQ-7000 platform is 2.022w. However, the power consumption of the PC platform is no less than 300w. Compared with the PC platform, it is no doubt that the energy consumption of the FPGA platform can be ignored. The real-time performance of the panoramic camera system and the panoramic stitching quality meet the design requirements.

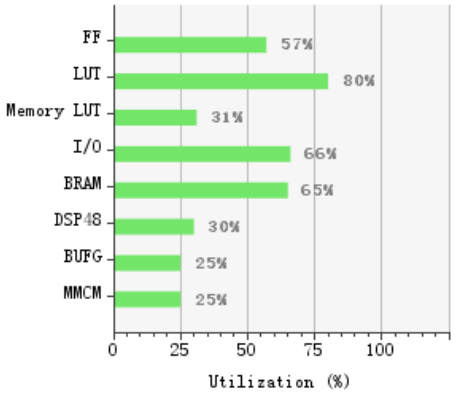


Fig. 27. The utilization of the panoramic stitching system.

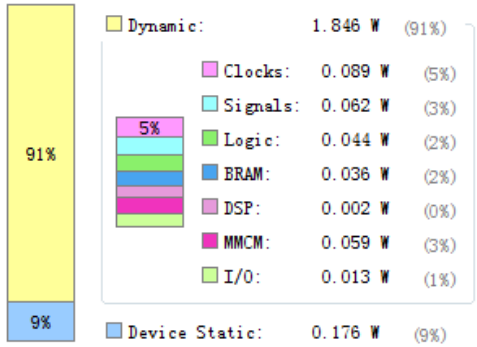


Fig. 28. The power consumption of the panoramic stitching system.

In recent years, many researchers have the strong interest in this field. Current panoramic technology [22] [23] [24] [25] focused on CPU speed-up, but it still cannot satisfy the requirement of real-time processing. Compared with existing 360-degree panoramic cameras [26][27], the proposed panoramic stitching algorithm has a relatively small distortion and a good stitching quality. Since deformation of the 360-degree panoramic lens is relatively large, so it is difficult to solve the image distortion problem.

There is no doubt that the cylindrical panoramic generation method based on multi-cameras has its unique advantages. Taking advantages of FPGA + ARM architecture, we can get a faster processing speed and obtain a better panoramic stitching result than a PC platform.

V. Conclusion

This paper presented a high-speed 360-degree panoramic video acquisition system based on FPGA-based parallel processing and USB 3.0 interface. One of the important features is that the image stitching is conducted on the FPGA platform, so real-time panoramic images can be captured at a speed of 60 fps. The SIFT features are used to register and fuse two adjacent images in order to improve the robustness. A FPGA-based implementation of the SIFT extraction algorithm has been presented. Compared to those based on PC, the developed SoC solution leads to a compact stand-alone camera system with real-time and robust performance and low cost. The developed system has many potential applications in computer vision, robotics, environmental monitoring and industrial automation.

Acknowledgment

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

References

[1] N. Chiba and H. Kano, “Feature-Based Image Mosaicking,” *Systems and Computers*, vol.31, no.7, pp.1581-1589, 2000.

[2] V. C. S. Chew and F.-L. Lian, “Panorama stitching using overlap area weighted image plane projection and dynamic programming for visual localization,” in Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron. (AIM), Jul. 2012, pp. 250–255.

[3] B. S. Kim, S. H. Lee, and N. I. Cho, “Real-time panorama image synthesis by fast camera pose estimation,” in Proc. Asia-Pacific Signal, Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC), Dec. 2012, pp. 1–4.

[4] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, “Real-time panoramic mapping and tracking on mobile phones,” in Proc. IEEE Virtual Reality Conf. (VR), Mar. 2010, pp. 211–218.

[5] J.-H. Chen and C.-M. Huang, “Image stitching on the unmanned air vehicle in the indoor environment,” in Proc. SICE Annu. Conf. (SICE), Aug. 2012, pp. 402–406.

[6] H.-K. Kim, K.-W. Lee, J.-Y. Jung, S.-W. Jung, and S.-J. Ko, “A content-aware image stitching algorithm for mobile multimedia devices,” *IEEE Trans. Consum. Electron*., vol. 57, no. 4, pp. 1875–1882,Nov. 2011.

[7] D. Ghosh and N. Kaabouch, “A survey on image mosaicing techniques,” *Journal of Visual Communication and Image Representation*, vol.34, pp. 1-11, 2016.

[8] 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.

[9] 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.

[10] G. Michael, G. Helmut, and B. Horst, “Fast approximated SIFT,” *Proc. ACCV*, pp.918–927, 2006.

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

[12] 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.

[13] J. Zhu and M.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.

[14] X. Xie et al. “A study on fast SIFT image mosaic algorithm based on compressed sensing and wavelet transform”, *J Ambient Intell Human Comput*, vol 6, pp. 835–843, 2015.

[15] B. Ronak and S. A. Fahmy. “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.

[16] J. Herbert, S. Wilson, “FPGA implementation of a high-speed, real-time, windowed standard deviation filter,” *Electronics Letters*, vol.34, pp. 22–23,2016.

[17] M. Qasaimeh and 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.

[18] J. H. Wang and S. Zhong, “An Embedded System-on-Chip Architecture for Real-time Visual Detection and Matching,” *IEEE Transaction on circuits and systems for video technology*, vol.24, no.3, pp.525-538, march 2014.

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

[20] V. Bonato and 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.

[21] Y. Ke and R. Sukthankar, “A more distinctive representation for local image descriptors,” *Proc. Computer Vision and Pattern Recognition*, pp.506–513, 2004.

[22] Z. W. Ren and M. Fang, "A Parallel Strategy for Stabilization Algorithm of Panoramic Camera Based on Multi-CCD," 2016 12th World Congress on Intelligent Control and Automation (WCICA), June 12-15, 2016, Guilin, China, pp. 840-844.
[23] H. K. Jeon and J. M. Jeong, "An implementation of the real-time panoramic image stitching using ORB and PROSAC," 12th International SoC Design Conference November 02- 05, 2015, Gyungju, South Korea, pp. 91-92.
[24] M. X. Lin and G. Xu," Cylindrical Panoramic Image Stitching Method Based On Multi-cameras," The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems June 8-12, 2015, Shenyang, China, pp. 1091-1096.
[25] Y. G. Xiong and K. Pulli, "Fast Panorama Stitching for High-Quality Panoramic Images on Mobile Phones," IEEE Transactions on Consumer Electronics, Vol. 56, No. 2, May 2010, pp. 298-306.
[26] D. P. Lan and J. H. Deng, "A Fisheye Lens 360 Degree Panoramic Monitoring System Based on the FPGA," Proceedings of 2013 FPGA Workshop and Design Contest, southeast university, Nanjing, China, November, 2013, pp.1-7.
[27] G. I. Kweon and Y. H. Choi, "Image-processing Based Panoramic Camera Employing Single Fisheye Lens," Journal of the Optical Society of Korea, September 2010, Vol. 14, No. 3, pp. 245-259.

Cong-Yi Lyu received his B.E. degree in Automation from Henan Polytechnic University, Jiaozuo, Henan, China, in 2009 and the Master degree in Health technology engineering from Hongkong polytechnic University, Hongkong, in 2011.

He is working toward a Ph.D. degree at the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China. His research interests include object tracking, FPGA based image processing.

Jian-Qing Peng received a B.E. degree in mechatronics engineering from Huazhong Agricultural University, Wuhan, Hubei, China, in 2012 and an M.E. degree in mechatronics engineering from Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China, in 2014.

He is working toward a Ph.D. degree at the Department of Mechatronics and Automation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. His research interests include space robotics, SLAM, FPGA and image processing.

Wei-Guo Zhou received his B.E.E. and M.S.E.E. degrees in Mechanical Engineering from HeFei University of Technology, and Harbin Institute of Technology, China, in 2012 and 2014, respectively. He is currently pursuing his Ph.D. degree in Mechanical Engineering at Harbin Institute of Technology, China.

His research interests include FPGA implementation, 3D time-of-flight range imaging, Human Interaction, Machine Vision and Neural Networks.

Shan-Shan Yang received her B.E.E. and M.S.E.E. degrees in Mechanical Engineering from Shandong University, and Harbin Institute of Technology, China, in 2012 and 2015, respectively. She is currently an engineer in Mechanical Engineering at Harbin Institute of Technology, China.

Yun-Hui Liu received the B.Eng. degree in applied dynamics from the Beijing Institute of Technology, Beijing, China, in 1985; the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, in 1989; and the Ph.D. degree in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1992.

He was with the Electrotechnical Laboratory,Ministry of International Trade and Industry, Ibaraki, Japan, from 1992 to 1995. Since February 1995, he has been with the Chinese University of Hong Kong (CUHK), Shatin, N.T., Hong Kong, where he is currently a Professor with the Department of Mechanical and Automation Engineering and the Director of the CUHK T Stone Robotics Institute. He is also an adjunct professor at the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, Harbin, China, and the Director of Joint Centre for Intelligent Sensing and Systems, National University of Defense Technology, Hunan, China, and CUHK. He has published more than 200 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, medical robotics, multifingered robot hands, mobile robots, sensor networks, and machine intelligence. Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He is the Editor-in-Chief of Robotics and Biomimetics and an Editor of Advanced Robotics. He served as an Associate Editor of the IEEE TRANSACTION ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. He is an IEEE Fellow.