15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

FPGA Implementation of High Frame Rate and Ultra-Low Delay Vision System with Local and Global Parallel based Matching

Tingting Hu, Takeshi Ikenaga
Graduate School of Information, Production and Systems, Waseda University
Fukuoka, Japan
hutingting@fuji.waseda.jp

Abstract

High frame rate and ultra-low delay image processing system plays an increasingly important role in human-machine interactive applications which call for a better experience. Current works based on vision chip target on video with simple patterns or simple shapes in order to get a higher speed, while a more complicated system is required for real-life applications. This paper proposes a BRIEF based matching system with high frame rate and ultra-low delay for specific object tracking, implemented on FPGA board. Local parallel and global pipeline based matching and 4-1-4 thread transformation are proposed for the implementation of this system. Local parallel and global pipeline based matching is proposed for high-speed matching. And 4-1-4 thread transformation is proposed to reduce the enormous resource cost caused by highly paralleled and pipelined structure. In a broader framework, the proposed image processing system is made parallelized and pipelined for a high throughput which can meet the high frame rate and ultra-low delay system's demand. Evaluation results show that the proposed image processing core can work at 1306fps and 0.808ms delay with the resolution of 640×480. System using the image processing core and a camera with 784fps frame rate and 640×480 resolution is designed.

1 Introduction

Nowadays, human-machine interactive applications call for higher frame rate and lower delay for a better experience, such as gesture recognition, automatic driving [1], projection mapping [2], and so on. High frame rate and ultra-low delay is quite important for a better human-machine interactive experience. A related research [3] based on vision chip, created a high-speed dynamic projection mapping system which could work at 500fps with the delay of 6ms. However, it cannot work on complicated objects and the 6ms delay is not competent for the higher frame rate situations while daily life requires a more complicated system which can deal with practical problems. FPGA is a good choice for dealing with complicated situations and could process with high speed under appropriate design. Therefore, FPGA implementation based high frame rate and ultra-low delay matching system comes to life.

BRIEF [4] algorithm which stands for Binary Robust Independent Elementary Features, can directly build short descriptors by comparing the intensities of pairs of points. Compared with SIFT (Scale-Invariant Feature Transform) [5,6], SURF (Speeded Up Robust Features) [7] and other conventional local features, BRIEF has a huge advantage over computation complexity, which is quite suitable for hardware implementation and causes lower delay.

Although BRIEF brings a great convenience for hardware implementation because of its binary characteristic, there are still some problems for high frame rate and ultra-low delay implementation on FPGA. Firstly, compared with conventional local descriptors, BRIEF generate descriptors in a much larger area around keypoints, which brings longer delay when generating such large block. Secondly, there's a great challenge for real-time hardware implementation of the matching part. The commonly used matching method in software is not hardware friendly for real-time implementation and it's difficult to finish the matching process with a short delay. The conventional work [8] constructs a SIFT based matching system which can work at 60fps. It uses template image as the query image, and a threshold is used to eliminate the unmatching pairs, which is hardware friendly. However, descriptors of the template are stored in a block RAM and the train descriptor needs to visit the template descriptors one by one because of the limitation of the block RAM, which takes much time. Finally, large resource cost is generated when the whole system is made parallelized and pipelined for a high throughput.

Aiming at high frame rate and ultra-low delay, this paper proposes a BRIEF based matching system and its real-time hardware implementation for VGA resolution video. First, the learning method proposed in ORB (Oriented FAST and Rotated BRIEF) [9] is used to choose a good subset of binary tests in a relatively small area around keypoints, which could reduce delay caused by line buffer. Local parallel and global pipeline based matching is utilized to accelerate the matching part which takes an important position in the matching system. And the heavy resource cost caused by the highly parallelled and pipelined structure is solved by 4-1-4 thread transformation. Next, the overall is parallelized into four threads in our hardware structure and the kernels processed in each module are pipelined. In this way, matching system with high frame rate and ultra-low delay is designed with less resource cost.

2 Proposed FPGA implementation of high frame rate and ultra-low delay matching system

This section shows the proposed BRIEF based matching structure of high frame rate and ultra-low delay vision system. The proposed structure of the matching system is shown in Fig. 1. The whole matching system is paralleled into 4 threads and the kernels processed in each module are pipelined. First, we uti-

lize Harris detector to extract the feature points. Then, Gaussian filter is used to smooth the input grayscale image data. Before generating descriptor of the feature point, 4 threads is transformed into 1 thread in order to reduce the enormous resource cost caused by highly paralleled and pipelined structure. Next, learning based binary tests subset selection is used to generate descriptor of the feature point in a block with a relatively small size. After obtaining the descriptor of the feature point, Local parallel and global pipeline based matching is proposed to find the feature which is matched with it from the template within lower delay. Finally, we transform the 1 thread into 4 threads and output the matching information.

Subsection 2.1 is about using the learning based binary-test subset selection to reduce the delay caused by line buffer. In subsection 2.2, local parallel and global pipeline based matching method is introduced to accelerate the matching process. In subsection 2.3, 4-1-4 thread transformation is described as to reduce the enormous resource cost caused by highly parallelled and pipelined structure.

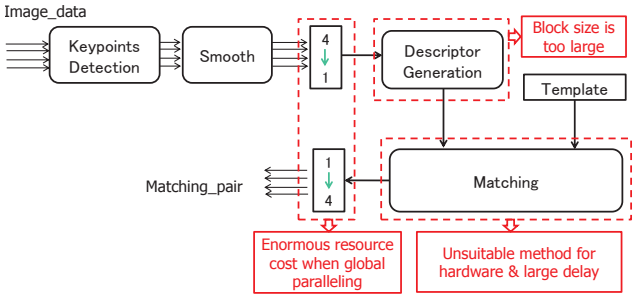

Figure 1. Proposed structure of matching system.

2.1 Learning based binary tests subset selection

Considering hardware implementation, BRIEF is used to generate descriptors because of its binary characteristic. BRIEF generates each bit of binary string according to the comparison of selected pixel pair inside the patch around keypoint. Generally, a typical BRIEF descriptor is made of 128, 256 or 512 comparisons. The patch size is 48 of length, and Gaussian distribution is utilized to select the subset of binary tests. It's possible to generate more distinctive descriptors with a larger patch, while larger resource consumption will be brought when generating such larger block on hardware. In block generation, we get the first block information after the first pixel of a frame comes to the center of the patch. When the patch size is 48×48, at least 24-line delay is brought by the block generation and a large amount of Flip Flops are needed, which is heavy for high frame rate and ultra-low delay system. Balancing the performance and delay caused by the large block, learning based binary tests selection is utilized to solve this problem.

ORB proposes an unsupervised learning method to select binary tests which have high variance and are uncorrelated by searching among all possible binary tests. Considering hardware friendly implementation, we use the learning method proposed by ORB to select binary tests in the patch with the size of 31×31,

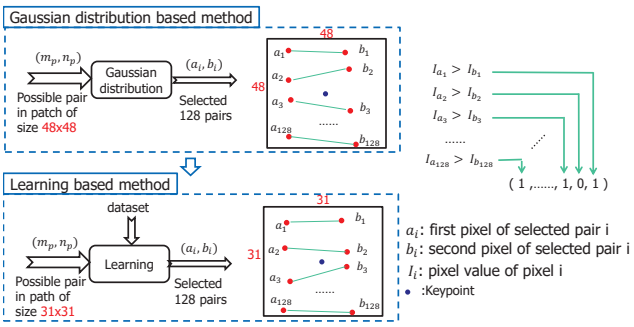

Figure 2. Concept of learning based binary tests selection.

and the first 128 binary tests with larger variance and less correlation are selected as the pattern to generate a 128-bit descriptor. And the process for descriptor generation is shown in Fig. 2.

2.2 Local parallel and global pipeline based matching

In the matching part, the commonly used matching method is Brute-force descriptor matcher, which finds the closest descriptor in a query-descriptor set by trying each one. Although we can optimize the algorithm to be much more hardware-friendly by using the template as the query image and preparing a threshold to eliminate the unmatching pairs, how to finish the matching process with a short delay is still a problem for high frame rate and ultra-low delay system. In order to meet the demand for high frame rate and ultra-low delay system, Local parallel and global pipeline based matching method is proposed to accelerate the matching process.

In general, we store the descriptors of template in a block RAM and read out the descriptors one by one to calculate the distance between it with the current processing descriptor. We can only do sequential processing when finding the matching point with the current processing descriptor from the template. However, the bit width of BRIEF descriptor is quite short, which means it's possible to save all the descriptors of the template in a register. We can read out all the data of the template at a time when data is saved in a register, which makes it easy to do parallel processing. Therefore, we propose Local parallel and global pipeline based matching and try to get matching information for each pixel within lower delay.

The structure of Local parallel and global pipeline based matching is shown in Fig. 3. Hamming distance is utilized to evaluate the descriptor similarity because of the characteristic of the binary descriptor, and 64 descriptors selected from the template are stored in a template register. There are mainly 3 parts in the matching process. First, we do 'XOR' operation between the current train descriptor and all the descriptors in the template register simultaneously, and 64×128-bit xor data can be obtained. Then, the number of '1' for each xor data which represent the distance between the current train descriptor and each query descriptor needs to be counted. Considering maximum frequency of the system, 64×128-bit xor data is divided into 4 parts and every 64×32 xor data is processed

simultaneously. The minimum distance can be found after obtaining all the distance between the train descriptor and all the query descriptors, and a distance threshold is prepared to eliminate unmatching pairs. The whole matching process is made fully parallelled and pipelined. In this way, matching module with high speed is designed.

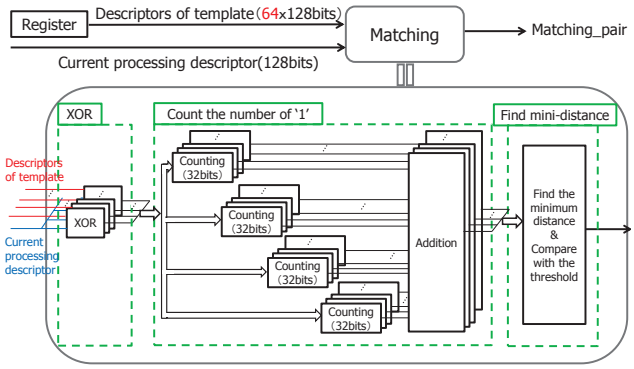

Figure 3. Structure of Local parallel and global pipeline based matching.

2.3 Local maximum neighboring check based 4-1-4 thread transformation

The whole matching system is parallelized into 4 threads in our hardware structure in order to get a high throughput which could meet the demand for high frame rate and ultra low delay system. However, the descriptor generation part and the matching part take lots of resource because of the local parallel processing, and there will be a great usage of resource when both the descriptor generation part and the matching part are parallelized into 4 threads. Therefore, 4-1-4 thread transformation is proposed to reduce the resource cost at the same time ensuring the throughput.

Our method is inspired by a general method - Local maximum neighboring check, which is used to avoid sorting during the feature extraction process with Harris. According to Local maximum neighboring, a pixel is a feature point only when its Harris value is bigger than other values in its surrounding block. Considering this idea, we think that there will be only 1 feature in simultaneously processed 4 pixels if the block size of neighboring check is 7×7. And actually, only feature point needs to be processed in descriptor generation part and matching part. Therefore we consider a 4-1-4 thread transformation method to reduce the large resource cost caused by the highly parallel and pipeline structure.

As shown in Fig. 4, we consider transforming the 4 threads into 1 thread at the beginning of descriptor generation. We generate descriptor and do matching process for the pixel which is a feature point. At last, transform the 1 thread into 4 threads at the end of matching. In this way, resource cost in descriptor generation part and matching part decreased by 1/4.

3 Hardware structure

As Fig. 5 shows, the input of our system is a video stream and the grayscale data comes in 10 pixels at a

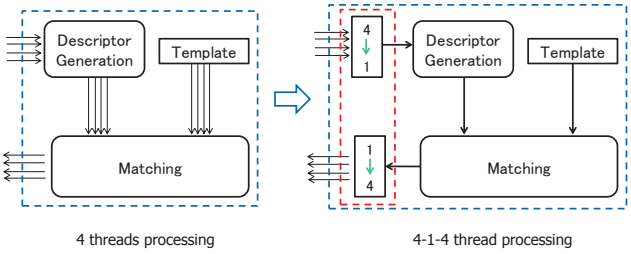

Figure 4. Concept of 4-1-4 thread transformation.

time. The data is transformed into 4 paralleled data through Camera link receiver. Then the 4 paralleled data is input into image processing module. The data goes through Sobel, Harris and neighboring check, descriptor generation and matching modules. Finally, a 32-bit data output from the image processing module is generated. 16-bit data of the output includes image information, feature information, and matching information. The extra 16 bits can store additional information for further use. Among all these modules, there is a memory controller which could control to write the data output from the image processing module into DDR3-SDRAM, as well as read out data from DDR3-SDRAM. And also, PC and FPGA could communicate with each other through WISHBONE-BUS. The data transform allows us to adjust the parameters by PC, send template data from PC to FPGA and do some post-process on the computer.

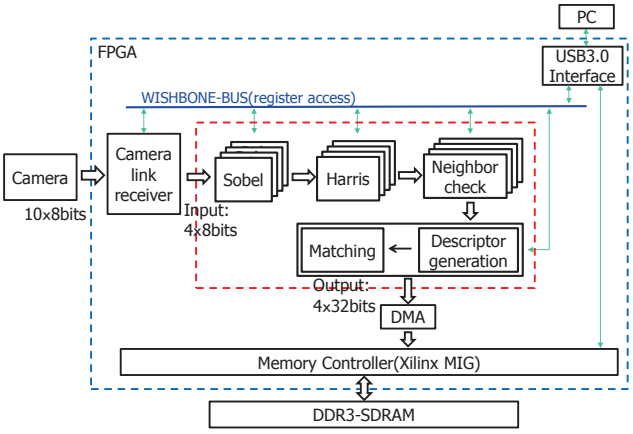

Figure 5. Hardware structure of the whole matching system.

A demonstration of our system is shown in Fig. 6. The high-speed camera we use is BASLER acA2000-340, which could capture 640×480 video with a frame rate of 784fps. And the FPGA board we use to process the captured video is Xilinx Kintex-7 XC7K325T. The information processed by FPGA board is sent to PC for a further processing.

4 Evaluation results

4.1 Matching performance

We use the Oxford dataset [10] to evaluate the performance of our method. Also we compare the per-

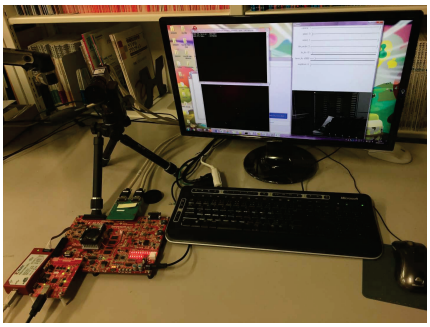

Figure 6. Demonstration of matching system.

formance with BRIEF and ORB. Here the BRIEF is evaluated by combining Harris detection and BRIEF. The experimental results on matching score is shown in Table 1. And larger matching score indicates better matching performance. As can be seen from Table 1, our method can reach almost the same performance with original BRIEF.

Table 1. Matching performance

	Bikes	Trees	Leuven	UBC	Wall
ORB	56.2	23.6	38.6	76.8	16.4
BRIEF	73.6	32.2	68.2	89.8	35.4
our method	67.2	27	67	88.8	37

4.2 Hardware performance

The whole system of FPGA can work at a maximum frequency of 100.331MHz. The process time for 1 frame in the image processing part is 0.808ms. And 42.699μs is needed for a pixel to travel through the image processing part. In theory, the designed image processing part can support 640×480-video processing with a frame rate of 1306fps. Actually, it's needed to consider exposure time and the time when reading image from the camera. And this system can process VGA video with a maximum frame rate of 784fps when connecting with camera BASLER acA2000-340.

4.3 Hardware resource usage

Table 2 shows the resource usage of this system. We can see this system doesn't require even half of the total resource, which means we can implement other modules to improve the performance of this system.

5 Conclusions

In this paper, a high frame rate and ultra-low delay matching system is proposed. The whole system is implemented on the FPGA board with input from a high frame rate camera and PC.

This work aims at a practical high-speed matching system that can be contributed to object tracking which can deal with real-life events. This paper mainly proposes 2 ideas to get the work completed. Firstly, local parallel and global pipeline matching is used to accelerate the matching part. Secondly, based on local maximum neighboring check, 4-1-4 thread transformation is proposed to reduce the large resource cost caused by the highly parallelled and pipelined structure.

As a result, the proposed image processing core can work at 1306fps and 0.808ms delay with the resolution of 640×480.

Table 2. Resource usage of proposed design

	Used	Total	Percentage
♯ of slice registers	60,406	407,600	14%
♯ of slice LUTs	68,841	203,800	33%
♯ of occupied slices	22,481	50,950	44%
♯ of bounded IOBs	225	500	45%
♯ of BUFG /BUFGTRLs	10	32	31%
♯ of DSP48E1s	28	840	3%

References

[1] M. Heesen, M. Dziennus, T. Hesse, M. R. Baumann.: "Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling," *IET Intelligent Transport Systems*, vol.9, pp.95–104, 2015.

[2] J. Chen, T. Yamamoto, T. Aoyama, T. Takaki, I. Ishii.: "Simultaneous Projection Mapping Using High-frame-rate Depth Vision," *IEEE International Conference on Robotics and Automation*, 2014.

[3] Y. Watanabe, G. Narita, S. Tatsuno, T. Yuasa, K. Sumino, M. Ishikawa.: "High-speed 8-bit Image Projector at 1000 fps with 3 ms Delay," *The International Display Workshops*, 2015.

[4] M. Calonder, V. Lepetit, C. Strecha, P. Fua.: "BRIEF: Binary Robust Independent Elementary Features," *European Conference on Computer Vision*, 2010.

[5] D. G. Lowe.: "Distinctive image features from scale-invariant key points," *Internatinal Journal of Computer Vision*, vol.60, pp.91–110, 2004.

[6] D. G. Lowe.: "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, 1993.

[7] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool.: "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol.110, pp.346–359, 2008.

[8] T. Suzuki, T. Ikenaga.: "Low Complexity Keypoint Extraction Based on SIFT Descriptor and Its Hardware Implementation for Full-HD 60fps Video," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol.E96-A, pp.1376–1383, 2013.

[9] E. Rublee, V. Rabaud, K. Konolige, G. Bradski.: "ORB: an efficient alternative to SIFT or SURF," *IEEE International Conference on Computer Vision*, 2011.

[10] K. Mikolajczyk, C. Schmid.: "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.27, pp.1615–1630, 2005.