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

Quadrant Segmentation and Ring-like Searching based FPGA Implementation of ORB matching system for Full-HD video

Tianmin Rao, Takeshi Ikenaga
The Graduate School of Information, Production and Systems, Waseda University
Kitakyushu, Japan
raotm@ruri.waseda.jp

Abstract

Full-HD video has drawn more and more attention in advanced computer vision applications which rely on more details in image. Benefit from high resolution input, local feature based matching system which at base of various vision applications, can also get better performance due to more available information. However, high resolution brings massive data and makes it challenging to achieve real-time and low cost at the same time. This paper proposes an ORB-based matching system for Full-HD video which implemented on FPGA. To improve nonlinear functions and feature steering part of ORB in hardware, the Quadrant Segmentation based orientation detector and Ring-like Searching based feature steering are proposed to make original operation more suitable for hardware. Evaluation shows that the proposed ORB matching system can complete feature extraction and matching for one Full-HD(1920×1080) image within 13.37ms and save almost 75% resources on average in feature extraction part compared with SIFT-based design.

1 Introduction

During these years, Full-HD video based applications gradually shown its potential in many computer vision application such as stereo matching and depth map[1], face-detection [2]. For these applications, high resolution image can provide more details than low resolution image and makes a better performance. Local feature based matching is at base of various vision system and by obtaining more information from image, it can also improve the ability of matching details.

However, high resolution contains massive media-data and it is challenging for applications which rely on real-time. Conventional work uses SIFT-based matching system to achieve real-time for Full-HD video [3], but the generation and utilization of SIFT descriptor is still complex. Recently, Oriented Fast and Rotated BRIEF (ORB) [4] are widely used in real-time oriented applications. The binary based data structure of ORB allows it to operate data with fewer dimensions and make a enormous difference when deal with a large database. ORB also robust to image rotation so that applications with rotated viewpoint can also obtain a satisfied performance.

The remained weakness of applying ORB to real-time hardware system lies in two aspects. First, ORB's orientation detector contains some nonlinear functions and nonlinear processing is inefficient for hardware since hardware mainly build on logic operation. Second, the feature generation part need randomly access to memory for getting pixel value. For hardware, this operation would cost much clock cycle and lower down the whole processing speed.

In this paper, our work mainly focus on achieving a low-cost hardware design for ORB feature generation to replace the previous SIFT feature based design. To overcome the weakness of hardware implementation for ORB features, this work proposed a Quadrant Segmentation based orientation detector and a Ring-like searching based feature steering. The first method assign orientations for an image patch by geometry judgement using a LUT of 36 fixed directions. And the second method changes the original rectangle pixel patch into Ring-like region which aims at simplifying the data structure for data accessing.

The rest of this paper is organized as follow. Section 2 discuss challenges in hardware implementation of ORB. Section 3 shows the proposed hardware structure and details about our proposed modules. Section 4 gives evaluation results and Section 5 contains conclusion to this paper.

2 Challenges in hardware implementation of ORB

ORB is a binary based local feature which robust to rotation. The algorithm uses FAST [5] to detect keypoint and our work focus on descriptor generation part. The computation flow of ORB is showed in Fig.1.

For the orientation detector, ORB uses the intensity centroid [6] to calculate orientation of the image patch. The intensity of a patch is defined as the offset from its center and described as moments:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{1}$$

Then the centroid C is calculated and the orientation θ is determined by the vector from patch center

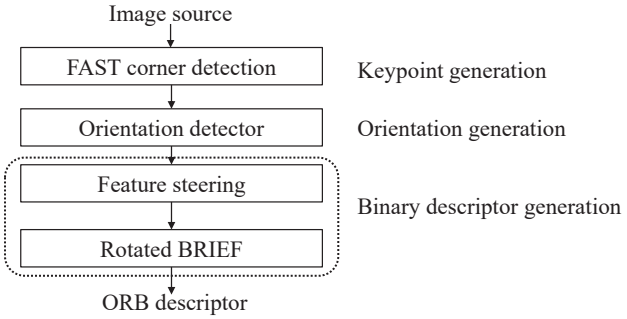


Figure 1. Computation flow of ORB algorithm.

O to C. The calculation of intensity centroid C and orientation θ is defined as:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}), \theta = \arctan 2(m_{01}, m_{10}) \quad (2)$$

A tough task for hardware is to calculate nonlinear function $\arctan 2(y, x)$. In software, this function firstly calculate the division of $c = y \setminus x$ and then use the mathematical method of *Taylor Expansion* to calculate $\arctan(c)$ for θ.

However, in hardware, multiplication on float values in *Taylor Expansion* need additional multiplier and even we apply a single value based LUT for $\arctan(c)$, the division for c is still costly.

For the feature steering part. ORB's binary descriptor is build on BRIEF [7] feature which is a table of point pairs geometry for testing. A set of n tests S is showed as:

$$S = \begin{pmatrix} x_1, & \dots & , x_n \\ y_1, & \dots & , y_n \end{pmatrix} \quad (3)$$

ORB rebuild this feature list with 2D rotation by matrix R_θ. The rotated feature set S_θ is:

$$S_\theta = R_\theta S \quad (4)$$

At runtime, we need to compare the image pixel values based on S_θ, so that we can get binary feature with consideration of orientation. This requires random access to memory, and is a tough task for hardware and the calculation for matrix R_θ is also costly.

3 Proposed Hardware Structure based on Quadrant Segmentation and Ring-like Searching

In Section 3, we first show overall structure of proposed ORB matching system in Fig.2. The keypoint detection part introduces another scale-invariant Harris detection design which is originally working for a SIFT-based design [3] and Our proposed method mainly focus on ORB feature extraction part.

In this system, the keypoint detection and feature extraction are working in parallel. To make whole system into pipeline processing, a 1920×31×8bit FIFO is used as line buffer. Each module gets data from this buffer and generate image patch for their own.

In Orientation detector, the Quadrant Segmentation module replaces the arctan function. Then, the orientation result θ are send to Feature steering part to generate rotated feature set S_θ. Then, Ring-like Searching module would rebuild feature set according to θ. The rotated BRIEF part generates descriptor based on S_θ. If the image patch under processing is a keypoint labeled by Harris detection, the generated descriptor would be send to block RAM for matching with database.

The matching part calculates Hamming distance between descriptors in block RAM and database. The path delay is used to make all processing in pipeline and complete 1 descriptor per clock.

In subsection 3.1, we show details about the Quadrant Segmentation based orientation detector. In the subsection 3.2, the Ring-like Searching based patch steering is showing in details about simplifying the data structure for a faster pixel value accessing.

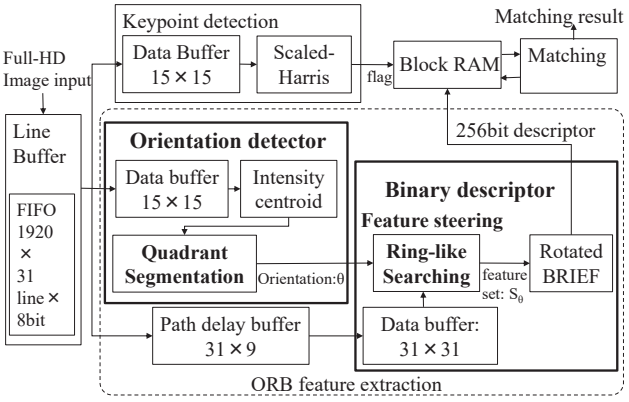


Figure 2. Overall hardware structure of proposed system.

3.1 Quadrant Segmentation based Orientation detector

In ORB, original orientation detector must calculate nonlinear function $\arctan(c)$. For conventional hardware method, single value based LUT is used for $\arctan(c)$ but the input c is a division result of float value and both division and float value is unfriendly for hardware.

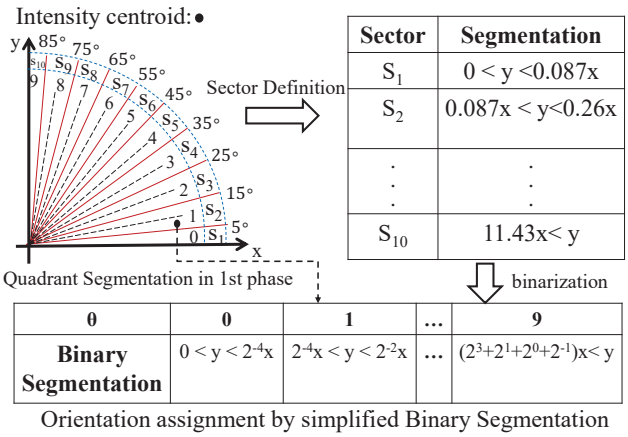
Another classical method is Coordinate Rotation Digital Computer(CORDIC) [8]. The key concept of CORDIC is using shift, addition and subtraction to gradually approach to the result by rotating the initiate vector but it costs several clock for iteration. Some well-designed CORDIC can work in pipeline but this would cost more resources and the accuracy is too high for our application.

Our method get inspiration from this idea and by combining geometry segmentation and a 2 stage LUT, we can directly assign patch rotation angle by its' intensity centroid coordinates C(x,y).

To assign orientation, the whole quadrant space are divided into 36 directions. The interval between each direction is 10 degree and the label from 0 to 35 are given to each. Then, each direction is assigned to a certain region which defined by segmentation lines and formed like sectors. Here we consider the first quadrant and show the segmentation and processing flow in Fig.3.

Segmentation lines start from 5° and end with 85° with the interval of 10°. The 10 directions are assigned into sector regions with label. Each line can be set according to linear function $y = kx$. The coefficients are set by $k = arctan(\alpha)$, α meanings the angle of segmentation lines. To refine this part for hardware, the coefficients k are transformed into the addition of binary shifting. We build the first LUT according to this and directly assign orientation.

For points in other quadrant, we determine the orientation by geometrical symmetry. For an intensity centroid C(x,y), we first detect which quadrant it belongs to and give it a label by checking sign bit. Then, we use the absolute value of x, y to assign orientation in 1st quadrant. Since arctan has symmetry in 4 quadrants, we further map the result by second LUT of Quadrant Mapping in Table1.



Orientation assignment by simplified Binary Segmentation

Figure 3. Quadrant Segmentation based orientation assignment in 1st phase.

Table 1. LUT for Quadrant Mapping.

Quadrant Label	Orientation Transform
0	0 + Label
1	18 - Label
2	18 + Label
3	36 - Label

The Label here from 0 to 35 represents the degree from 0° to 350°(the "36" are treat as "0"). By these two LUT, every point in the quadrant space can be mapped to a specific orientation. Using pipeline structure, this method can generate orientation for a image patch every clock.

3.2 Ring-like Searching based feature steering

In binary descriptor generation part, feature set S must be rebuild according to orientation θ. Original structure has problem in nonlinear function and randomly data accessing. We decide to skip them and arrange the data into Ring-like region for fast accessing.

The rotated feature set S_θ is a list of positions with 2d rotation for selected pixels. The rotated positions are calculated by matrix R_θ and then a searching in image patch is needed to getting the correct pixel value. Generally, this flow should utilize RAM to store pixel value and through addresses to randomly access to each. But this would cost much clock and lower down the processing speed.

In our design, since the orientation is fixed into 36 directions, the rotated feature set S_θ can be calculated before we start the processing. The rotated position for 36 directions can form a Ring-like region. The formation of one Ring-like region can be showed in Fig4.

Thus, the searching for pixel value only related to data in this Ring-like region. So if we can previously store them as a database, we can get the pixel value for each comparing according to the orientation of image patch. We build Ring-like database for binary test of 256 pairs in ORB's binary generation part. And for all these pairs, only 88 Ring-like databases are need.

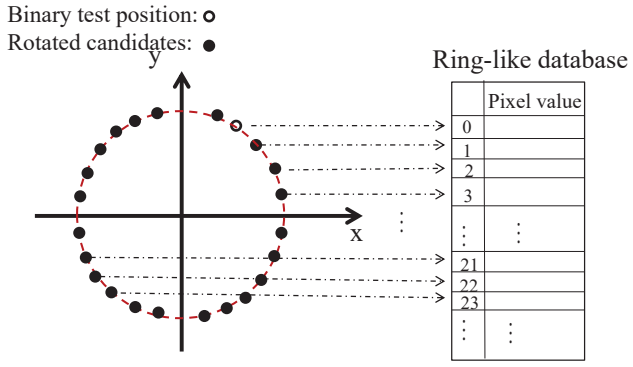


Figure 4. Ring-like database.

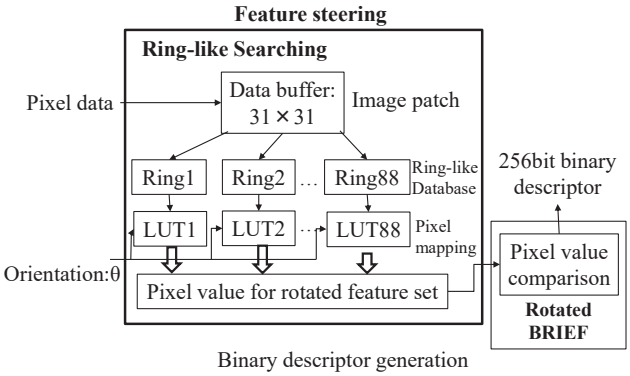


Figure 5. Hardware structure of Ring-like searching based feature steering and binary descriptor generation.

By this method, the expected pixel value can be directly accessed according to a specific orientation. From a "Ring", the pixel value can be extracted by LUT build on orientation of the image patch. The generation of binary descriptor can be done by operating comparison between each feature set and this is the original rotated BRIEF by ORB algorithm. The hardware structure of the proposed binary descriptor generation is showed in Fig5.

4 Evaluation result

In this part, we shows our evaluation result for hardware performance.

The software evaluation platform we use is vs2013 plus Opencv 2.4.13 and the program is in C++. The hardware evaluation environment we used is Virtex6(XC6VLX760) offered by Xilinx, Inc. The camera we use is HC-X900M which can provide Full-HD resolution video. The logic synthesis of FPGA design is performed by ISE 14.7.

4.1 Software performance

Our proposed method are approximately calculating for the orientation information, we use rotation deformation data set "boat" of Oxford dataset[9]. The matching of ORB descriptor is judged by hamming distance and it can show the similarity of matched feature.

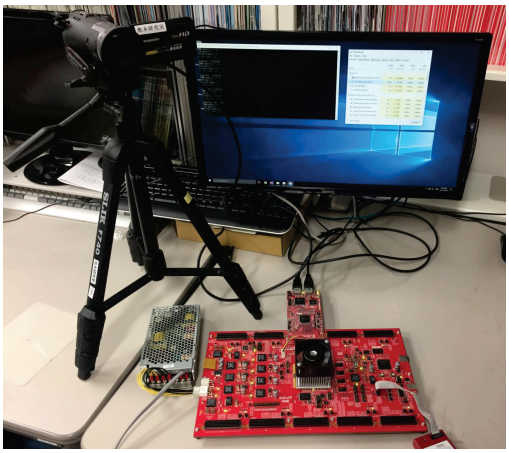


Figure 6. Demonstration Environment.

We calculate the rate of different hamming matches among 500 matched features to show difference by 4 kinds of deformation with rotation and zoom. The scale pyramid is from original ORB to only show the effect of our proposed rotation operator.

Table 2. Different rate between original ORB and proposed method

test label	test1	test2	test3	test4
Different matches	11%	12.8%	6.8%	5.2%

4.2 Hardware performance

The input video is in Full-HD (1920×1080) resolution with frame rate of 60fps. Our proposed matching system can complete processing of one frame about 13.37ms with 500 features. The result shows our proposed system can achieve real-time processing under the timing constrain of 16ms. The main parameter is showed in Table3.

Table 3. main parameter and system performance

Input video resolution	1920×1080
Input video frame rate(fps)	60
Maximum frequency(Mhz)	150.435
Processing time of one frame(ms) (for ORB feature extraction)	10.37
Processing time of one frame(ms) (for matching)	3.0
Total processing time of one frame(ms)	13.37

4.3 Hardware resources

We compare our proposed ORB feature generation part with a SIFT-based design [3] which also achieve real-time for Full-HD video. From the comparison, our proposed system can save almost 75% resources on average for orientation detector and descriptor generation. The comparison is showed in Table 4 and Table 5.

Table 4. Resources Comparison of orientation detector

	ORB	SIFT	Reduction
Slice Registers	5,468	25,637	78.6%
Slice LUTs	5,531	19,487	71.6%
Occupied Slices	2047	6,571	68.8%
BlockRAM/FIFO	23	0	0.0%

Table 5. Resources Comparison of descriptor generation

	ORB	SIFT	Reduction
Slice Registers	6,809	47,843	85.7%
Slice LUTs	15,770	88,245	82.1%
Occupied Slices	7,048	24,303	71.0%

5 Conclusion

This paper proposes a low-cost structure of ORB matching system for Full-HD video. The system implement FPGA and evaluation results shows it can work for real-time oriented applications. For resources, the proposed ORB-based design show advantage in low-cost of registers and LUT. By further optimization, this design can be used for other embedded vision system which aims at real-time and low cost.

References

[1] L. D. Chen, Y. L. Hsiao and C. T. Huang. "VLSI architecture design of weighted mode filter for Full-HD depth map upsampling at 30fps," *IEEE International Symposium on Circuits and Systems*, ISCAS, ,pp. 1578-1581, 2016.

[2] C. Oh, S. Yi and Y. Yi. "Real-time face detection in Full HD images exploiting both embedded CPU and GPU," *IEEE International Conference on Multimedia and Expo*, ICME, 2015, pp. 1-6.

[3] T. Suzuki and T. Ikenaga. "SIFT-based low complexity keypoint extraction and its real-time hardware implementation for full-HD video," *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1-6, 2012.

[4] E. Rublee, V. Rabaud, K. Konolige and G. Bradski. "ORB: An efficient alternative to SIFT or SURF," *International Conference on Computer Vision*, ICCV, pp. 2564-2571, 2011.

[5] Rosten E, Drummond T. "Machine learning for high-speed corner detection," *European conference on computer vision*, pp. 430-443), 2006.

[6] Rosin, Paul L. "Measuring corner properties," *Computer Vision and Image Understanding* vol.73, no.5, pp. 291-307, 1999.

[7] Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua. "Brief: Binary robust independent elementary features," *European conference on computer vision*, pp. 778-792, 2010.

[8] J. E. Volder. "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computers* vol.EC-8, no.3, pp. 330-334, 1959.

[9] `http://www.robots.ox.ac.uk/~vgg/research/affine/`