Advanced Science and Technology Letters
Vol.43 (Multimedia 2013), pp.58-62
http://dx.doi.org/10.14257/astl.2013.43.11

A hardware design of optimized ORB algorithm with reduced hardware cost

Kwang-yeob Lee[1], Kyung-jin Byun[2]

[1] Dept. of Computer Engineering, Seokyenog University,
Jeongneung 4-dong, Seongbuk-gu, Seoul, Korea
kylee@skuniv.ac.kr
[2] Multi-media Processor Research Team, ETRI, Daejun, Korea
kjbyun@etri.re.kr

Abstract. The conventional ORB algorithm is optimized for the hardware implementation. The optimized ORB algorithm is implemented on Zynq-7000 SoC Platform. SURF algorithm that is well known as a feature based algorithm is not suitable in the embedded environment, because it requires lots of calculations in the process. The proposed hardware accelerator reduces approximately 50% of internal memory usage and 25% of hardware logic compared to the conventional SURF accelerator, and it perform above 100 frames per second at 100MHz operation frequency.

Keywords: FAST corner detection, rBRIEF, intensity centroid, ORB algorithm, computer vision.

1 Introduction

Recently, studies on object recognition in computer vision for detecting the existence of an object in a certain range have been actively conducted in many applications. Scale Invariant Fourier Transform(SIFT)[1] and Speed Up Robust Features(SURF)[2] are widely used for object recognition. However, it is difficult to process real-time object recognition owing to their high computational complexity. To overcome this drawback, studies on designing SURF algorithms using a hardware accelerator are in progress. However, its disadvantage is that it incurs excessive hardware costs because using a hardware accelerator requires considerable internal memory.

In this paper, we propose to design ORB algorithm[3] as a hardware accelerator, instead of SIFT and SURF. The zynq-7000 SoC platform was used as an experimental setup for the analysis. The hardware accelerator proposed in this paper reduced logic slice usage by approximately 68% and block RAM usage by 83% compared to the SURF accelerator. The runtime was about 18 ms, which exceeds the performance of 100 frame per second.

ISSN: 2287-1233 ASTL

2. ORB algorithm

2.1 Feature point Detection

In the feature point detection stage, FAST corner detection algorithm [4] are used. The FAST corner detection algorithm is executed in the following order: feature detection (FD), feature score (FS), and non-maximum suppression (NMS). In the FD stage, the algorithm initially assigns an area of size 7 × 7 pixels around a central point, and then compares it to the surrounding brightness pixels. When compared to the brightness pixels above, below, and on both sides of the center point, if the sum is greater than 3, 16 nearby brightness pixels are compared. Then, if the sum of the comparison is greater than 12, they are detected as feature point candidates. In the FS stage, a score for each feature point candidate is determined based on threshold values. In the NMS stage, the scores of neighboring feature points are compared to the reference feature point and the largest or smallest feature point set as the final feature point.

2.2 Orientation

In the FAST corner detection stage, orientation is decided by an intensity centroid because not every feature point has an orientation component. The intensity centroid determines the orientation using a moment method. First, it assigns an area of 15 × 15 pixels around the feature points. Each feature point can be oriented using a primary moment and atan2 function.

2.3 Descriptor Generation

The rBRIEF algorithm is based on the BRIEF algorithm[5], which has a high descriptor generation speed. The rBRIEF algorithm is a bit string descriptor that consists of 0 and 1. After assigning an area of 31 × 31 pixels around the feature point and setting two nearby coordinates to p(x) and p(y), if p(x) is smaller than p(y), then the binary test becomes 1, else it becomes 0. By repeating the above process 256 times, a 256-bit string descriptor is generated.

3 Proposed hardware architecture

3.1 Hardware Cost Reduction with keypoint detection block

Fig. 1 shows the feature point detection block. In the source loader stage, the brightness image value is read sequentially from the external memory. In the FD stage, the feature point candidates are determined by comparing 16 brightness pixels around the central point at the same time. If a point is determined as a feature point

candidate, it is saved in FIFO memory. Then, in the FS stage, the feature point is assigned a score by reading the location of the point and by the surrounding brightness pixels. The NMS stage compares the score of the feature point to those of the surrounding feature points after reading the location and score of the feature point from the buffer. Further, if the score is greater than the score of the surrounding feature points, the feature point is detected as a final feature point.

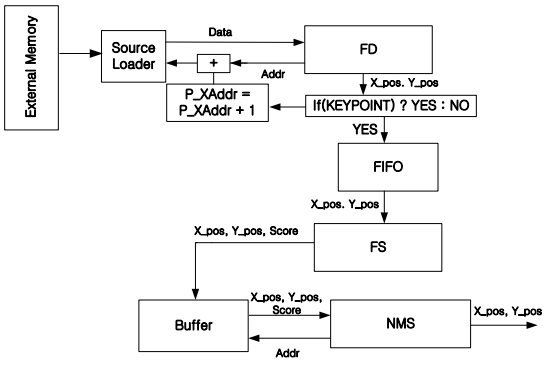

Fig. 1. keypoint detection block

3.2 Orientation Block

The execution process of the orientation block is as follows. The location of the feature point detected from the feature point detection block is saved temporarily in FIFO memory. In the orientation module, the primary moment is computed using the pixels around the location of the feature point. Atan2 LUT is a table created in advance to reference parts used in the atan2 function, and when this increases LUT usage, it improves performance.

3.3 Descriptor Generation Block

The execution process for the descriptor generation block is as follows. The feature point's location detected from the feature point detection block and the orientation component determined from the orientation stage is saved in FIFO memory. The rBRIEF module obtains the coordinates for comparison by reading coordinates from the cosine and sine data and the pattern lookup table saved in FIFO memory. By comparing the brightness pixels of the two points, a bit-string descriptor is created. In order to generate a 256-bit string descriptor, a coordinate should be read and compared 256 times. However, to enhance the performance, the algorithm is designed to compare eight coordinates in one cycle by adding internal memory. Hence, a 256-bit string descriptor can be generated in 32 cycles.

4 Experiment and Analysis

The zynq-7000 SoC platform was used as the experimental setup. Two images of graffiti and a sunflower with a resolution of 640 × 480 were used for image verification. In the experiment, 600 to 800 feature points were detected. Table 1 summarizes the results of ORB in a field-programmable gate array (FPGA) and embedded environment.

Table 1. comparison of computing times

	Embedded [3]	FPGA
Frequency	1 GHz	100 MHz
Exec. Time	66ms	18 ms

Table 2 summarizes the results of the ORB algorithm compared to those of the SURF hardware accelerator [6]. The slice logic and block RAM usage were reduced by approximately 68% and 83%, respectively. .

Table 2. comparison of hardware implementation cost and performance

	SURF [6]	Proposed ORB
Slice Logic	101,348	31,677
Block RAMs	752 KB	125 KB
Exec. Time	33 ms	18 ms

5 Conclusion

This paper introduced an ORB algorithm as a hardware component on the zynq-7000 SoC platform and compared the hardware costs and internal memory usage with the existing SURF hardware accelerator. The comparison demonstrated that the cost reduced by approximately 68% and the internal memory usage by approximately 83% when the ORB algorithm was used.

Acknowledgments

This work was supported by the IT R&D program of MOTIE/KEIT. [10035152 , Energy Scalable Vector Processor - Primary Technology]

References

1. D. G. Lowe, "Distinctive image features from scale-invariant keypoint", International Journal of Computer Vision, 60(2):91-110. 1,2.

2. H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features", European Conference on Computer Vision, 1:404-417, Graz, Austria, May 2006.
3. E. Rublee, V. Rabaud, K. Konolige, "ORB: An efficient alternative to SIFT or SURF", Computer Vision and 2011 IEEE International Conference, pp 2564-2571, Nov 2011.
4. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", In European Conference on Computer Vision, volume 1, 2006.1.
5. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Bi-nary robust independent elementary features", In European Conference on Computer Vision. 2010.
6. Na Eun-Soo, Jeong Yong-Jin, "FPGA Implementation of SURF-based Feature extraction and Descriptor generation", Journal of Korea Multimedia Society, Vol 16, No.4, April 2013.

