Advanced Science and Technology Letters
Vol.48 (CIA 2014), pp.42-46
http://dx.doi.org/10.14257/astl.2014.48.08

# Hardware Design of Feature Point Extraction using SIFT Algorithm

Gi-woong Shin[1], Jong-tae Sung[1], Young-Hyoung Kim[1] and Yong-hwan Lee [1,1,]

[1]Kumoh National Institute of Technology, 61 Daehak-ro (yangho-dong), Gumi, Gyeongbuk 730-701, Republic of Korea
{ssambak87, sjt, kic126, yhlee}@kumoh.ac.kr

**Abstract.** In this paper, we design hardware architecture to accelerate feature point extraction using SIFT(Scale Invariant Feature Transform) algorithm for real-time embedded applications. The process of SIFT needs excessive amount of computations and memory accesses. To solve this problem, changing computation order and FIFO buffer are adopted. As a result, this hardware can extract feature points on qVGA(320x240) image in real-time. The proposed hardware is implemented in FPGA.

**Keywords:** Feature point, SIFT, hardware, FPGA, DoG, image stitching

## 1 Introduction

Recently computer vision technology has been widely used for image processing like image recognition and object recognition on robot, mobile, security system and other fields. Especially, the object recognition [2-5], a subfield of image processing, has been mostly used for industry and military purpose on the past[1]. However, the object recognition is now applied to mobile devices and home appliances for face and gesture recognitions.

SIFT[6-7] extracts feature vector with images generated in various scale, which causes slowdown in computing speed. In this paper, to reduce computing time, SIFT is implemented in hardware. The hardware implementation of SIFT outperforms the software by parallelizing image processing.

---

1 Manuscript received July 1, 2012; revised August 1, 2012; accepted September 1, 2012.
Corresponding author : Yong-hwan Lee
This work (Grants No. C0095719 ) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013.

ISSN: 2287-1233 ASTL

## 2 Extracting feature point with SIFT algorithm

SIFT algorithm proposed by Lowe [6] is one of the local image featuring algorithms. Although the scale of the objects changes and rotates, its feature does not change easily. The performance of SIFT is better than other local featuring algorithms relatively[8]. SIFT algorithm has been used for object recognition, panorama, face recognition and extraction of feature points. SIFT algorithm is divided into 3 steps as follows : Gaussian image, difference of Gaussian and extracting the feature points.

### 2.1 Gaussian image

In Eq. 1, Gaussian images L(x,y) is generated by Gaussian filters from input images I(x,y). Gaussian kernel is shown as Eq. 2, where represents the scale of Gaussian filter, i is the scale index and S is the number of scale. In this paper, since we set the value of S to 1, the number of Gaussian images S+3 is four. is determined by which is the scale of the first Gaussian image. As shown Eq. 3, Gaussian image is made by filtering through the Gaussian filter having scale of , where the Gaussian image is made from the input image[9].

$$( ) ( ) ( ). \quad (1)$$

$$( \quad ) \quad \frac{}{} . \quad (2)$$

$$\sqrt{} \quad . \quad (3)$$

To extract feature points, SIFT algorithm makes Gaussian images by filtering masks that having different values of variance as shown in Fig. 1.

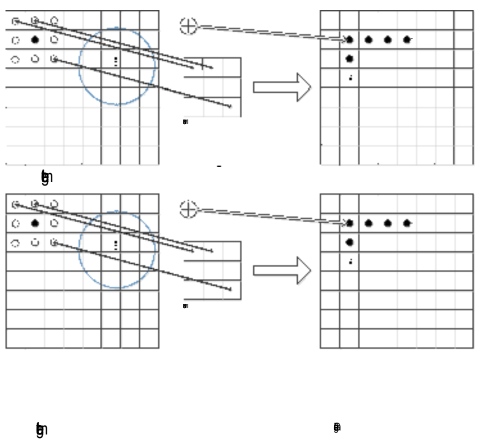



**Fig. 1.** Convolution in image process

### 2.2 DoG (Difference of Gaussan)

DOG D(x,y) is generated by difference between and . As a result, Three D(x,y)are generated from four L(x,y). Eq. 4 generates DOG image from two Gaussian images. As shown in Fig. 2, when the generated images have different variance, images are blurred differently at edge. After that, edge-based images are generated.

( ) ( ) ( ) (4)

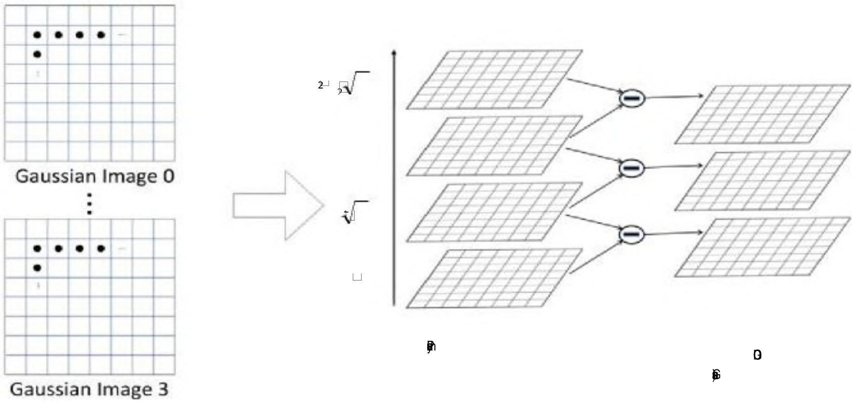


**Fig. 2.** Generating DOG

### 2.3 Feature Point Extraction

The feature point is extracted from Gaussian images. The each values of pixels are compared with twenty-six pixels in 3 3 3 window. If the value of pixel is the highest or the lowest among 3 3 3 window, this pixel could be a feature point.

## 3 Extracting feature point with SIFT algorithm

Gaussian

 

(Difference of

The implementation of SIFT by software is difficult to extract feature point in the real-time. It needs more memory and more computing time due to repetitive operations. Therefore, we propose hardware design to extract feature point in real-time. Adopted methods are following.

### 3.1 Procedure improvement of input image

The existing SIFT algorithm receives a data at a time and provokes waiting time for image data. So, the method that receives five data at a time is adopted.

### 3.2 Improvement of the time for data entry

The time spent in data entry is improved by using FIFO Buffer. The existing SIFT algorithm processes are performed in sequence. The input image is stored in FIFO buffer. The stored image is processed by the Gaussian filter in parallel. So, pre-DoG, post-DoG and delay line buffer are not required. The operation of the FIFO buffer is explained as follows. Input image is supplied as the size of Gaussian filter at the once, each of Gaussian filters accepts pixels as much as its own size. The direction of operation of the filter is performed in the vertical direction. If there is no input image contained in the FIFO Buffer, the operation of filter shifts to the right by one pixel vertically. This calculation is the result of an operation on a pixel point. So the delay time is reduced, whereas computing speed is faster. Hardware architectures to make the Gaussian image and to create a DOG are shown in Fig. 3.

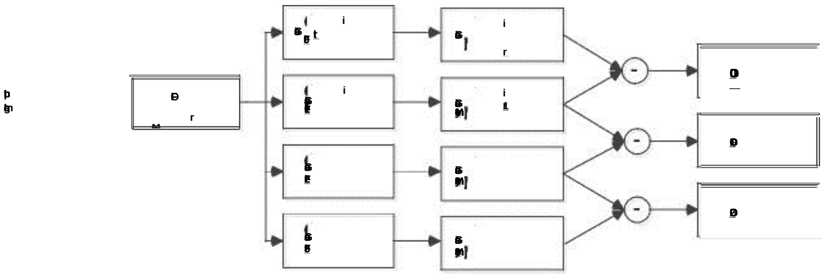

**Fig. 3.** Hardware structure of the module making Gaussian image

### 3.3 The hardware design for feature point extraction

In this paper, two lines of latest computed DoG are saved in two temporary buffers. These data are used to comprise 3 × 3 window for feature point extraction before explained. This method can avoid the duplication of calculating, thereby the computing process can be more efficient. The proposed hardware design for feature point extraction is shown in Fig. 4.

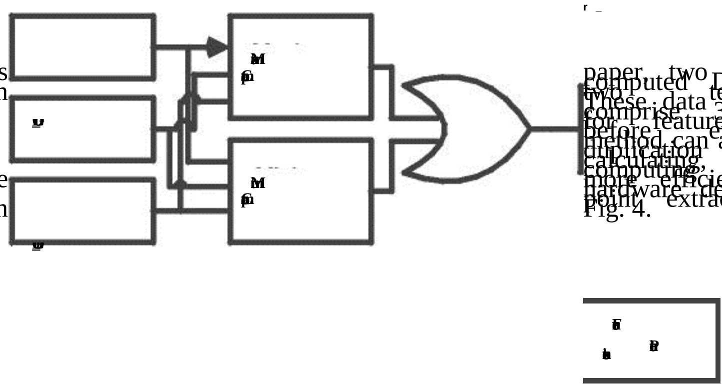

**Fig. 4.** Structure of feature point extraction hardware

## 4 Design verification and performance evaluation

In this paper, feature point extraction algorithm is verified through Verilog HDL and FPGA for functional verification and hardware verification. The implementation with parallelized hardware is tested by vertex4-XC4VLX60 of Xilinx with 206MHz clock. As a result, the performance is 111fps and acceptable for real-time system.

## 5 Conclusion

Real-time SIFT algorithm for feature point extraction is possible by using parallel processing architecture and FIFO buffer. The proven algorithm was modeled to RTL level using Verilog HDL and functional verification was verified by Modelsim. The SIFT algorithm for feature point extraction in real-time was synthesized by the XILINX ISE on the Virtex4. Since it is possible to operate in a real-time system, it can be used for object recognition, panoramic image, and mobile devices.

**Acknowledgments.** This work (Grants No. C0095719) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013.

## References

1. Treiber, S. M.: An Introduction to Object Recognition, Springer, pp. 1--20, 130--149 (2010).
2. Mundy, J.: Object Recognition in the Geometric Era: A Retrospective, lecture notes in computer science - toward category-level object, vol. 4170, pp. 3--28 (2006).
3. Viola, P. and Jones, M.: Robust real-time face detection, Int. J. Computer Vision, vol. 57, no. 2, pp. 132--165 (2004).
4. Zhang, W., Yu, B., Zelinsky, G.: Object class recognition using multiple layer boosting with heterogeneous features, in proc. IEEE CVPR, vol. 2, pp. 320--330. San Diego (2005).
5. Wang, C. and Wang, K.: Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction. Lecture Notes in Control and Information Sciences - Recent Progress in Robotics, vol. 370, pp. 300--330 (2008).
6. Lowe, D.: Object recognition from local scale-invariant features, in proc. of the Int. Conf. Computer Vision, Volume. 2, pp.1150--1157 (1999).
7. Lowe, D.: Distinctive Image Features from Scale-Invariant Key points, Int. Journal of Computer Vision, vol. 60, pp 91--110 (2004).
8. Mikolajczyk, K., Tuytelaars, T. and Schmid, C.: A Comparison of Affine Region Detectors, Int. J. Computer Vision, vol.65, pp.43--72 (2005).
9. Lindeberg, T.: Scale-space for discrete signals, IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 3, pp.234--254 (1990).

 