Adv Sist Sci Appl 2017; **17**(2); 43-51
Published online in http://ijassa.ipu.ru/ojs/ijassa/article/view/258

Reconfigurable Architecture for Image Feature Detection

Rajesh Nandalike[1], Saroja Devi Hande[2]

1) Department of Electronics and Communication Engineering

E-mail: nrajesh7@gmail.com

2) Department of Computer Science and Engineering

Nitte Meenakshi Institute of Technology, Bengaluru, India

E-mail: hsarojadevi@gmail.com

Abstract: Hardware-based developments are useful for variety of image based applications such as highly challenging video surveillance. FPGA comprises of combination of the hardware attributes of an ASIC, supporting reconfigurability, reduced time-to-market and real-time performance. Hardware-based feature detection is a promising solution that exploits inherent parallelism in algorithms to accomplish significant improvement in speed. The efficient usage of resources still remains a challenge that specifically determines the cost of hardware, which can be addressed using our approach. In this paper, we have presented the hardware architecture for feature detection part of the Scale Invariant Feature Transform (SIFT) algorithm for frames from an HD-720p video. The proposed architecture is designed using Xilinx System Generator (SysGen) tool and implemented on Genesys2 Kintex-7 FPGA Development Board.

Keywords: Field Programmable Gate Array (FPGA), Scale-Invariant Feature Transform (SIFT), Feature Detection, Keypoint Detection, Feature point, Image feature.

1. INTRODUCTION

Feature detection and matching are fundamental functionalities in many image applications, video surveillance and also computer vision. They are computationally intensive, requiring considerable amount of resources in terms of time, power, memory and silicon. Scale invariant feature transform (SIFT) algorithm proposed by Lowe is one of the potent algorithms in image matching and object recognition [1,2]. High resource utilization and computational complexity make SIFT algorithm challenging to cope up with the real-time performance for software implementation.

Feature-based identification is the prevailing object identification strategy. It employs one of the feature extraction algorithms to extract the important features of the image. The development of numerous feature extraction algorithms has been taking place during last few years. Canny and Sobel's edge detectors, Binary Robust Independent Elementary Feature (BRIEF) and Harris corner detectors are local feature extraction algorithms that are employed to extract features from an image in object recognition systems [3,4]. Every algorithm attempted to enhance the uniqueness and robustness across image transformation operations.

This paper proposes hardware architecture for feature detection on FPGA, so that the chip acts as a standalone system for feature detection. Based on the analysis of feature detection algorithms, five steps must be performed for feature detection and matching, visualization, Pre-processing, feature detection, feature descriptor building, feature matching and post-processing. SIFT algorithm is used for feature detection in this paper.

The approach aims at suitable algorithmic selection and testing using MATLAB. Different algorithms are tested and the corresponding hardware implementation architectures are investigated. Feature detection techniques are implemented on FPGA, which are directly adaptable by robotic units and air-borne vehicles.

2. SCALE INVARIANT FEATURE TRANSFORM (SIFT)

SIFT is the most competent approach to identify and describe invariant features of an image [5]. The features obtained are invariant to image scaling, rotation and partially invariant to change in illumination. It is an algorithm where image data is converted to scale-invariant coordinates corresponding to its local features.

SIFT consists of four major stages:

- Scale-space peak selection: In the first stage, the identification of feasible interest points in an image with their position and scale is accomplished. This is utilized effectively to find out the interest points that are stable, by constructing the Gaussian pyramid and searching for the interest points in a series of difference-of-Gaussian (DoG) images.
- Key point localization: In the second stage, the feature points are restricted to sub-pixel precision and are waived if they are ambiguous.
- Orientation assignment: In the third stage, orientations are allotted to each and every feature point position depending on the image gradient directions. The orientation, scale and location to each feature point empowers SIFT to construct an accepted perspective for the feature point that is invariable to identical transformations.
- Key point descriptor: In the fourth stage, image feature descriptor is built for each feature point. The SIFT algorithm forms a depiction for each feature point depending on a patch of pixels in its neighborhood. The patch that has been formerly centered about the feature point's location will be rotated depending on the dominant orientation and scaled to suitable size. The SIFT descriptor produces 128 feature vectors for each feature point [69].

3. FEATURE DETECTION

The properties of features found in an image make them relevant for identifying different images of a similar scene. Detecting good features is a complex problem. The following section gives a discussion on the properties of the ideal local feature. The desired properties from a feature depend on the actual application.

A feature is a point-of-interest in an image. It is a slice of information which is suitable for determining the computational assignment identified with respect to a particular application. The properties of a good feature are: they are consistent over several images of the same scene, insensitive to noise, invariant towards certain transformations [10].

SIFT uses scale-space extrema as candidate features. This work concentrates on the scale-space extrema detection with focus on dedicated hardware implementation.

The Gaussian kernel is well-defined in 2D as,

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

In equation (1), σ determines the width of kernel and is often referred as the inner scale. It is the standard deviation while the σ^2 is the variance. The term $1/(2\pi\sigma^2)$ is the normalization constant and it makes the integral over the exponential function unity. This constant in the equation makes it a normalized kernel with the integral unity for every σ. This means that increasing the σ value effectively decreases the height of kernel while increasing its width.

The procedure of SIFT feature point detection comprises of developing difference-of-Gaussian (DoG) pyramid of the image. From the developed DoG pyramid, maxima and minima known as scale-space extrema (also known as feature points or interest points) are recognized. The procedure of SIFT feature identification is given in fig. 1.

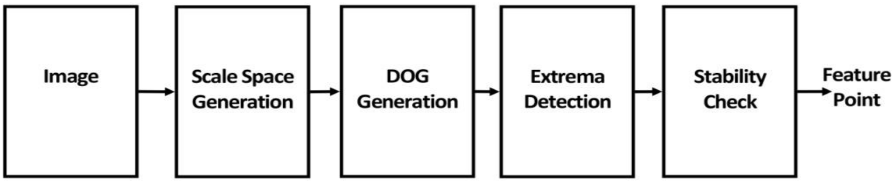


Fig. 1. Computational flow of Feature detection

3.1. DoG Pyramid Construction

The SIFT algorithm focuses on the image locations that display immense neighborhood transforms in their visual appearances. The DoG pyramid is built with the objective of identifying the feature points. The input image $I(x, y)$ is convolved with a Gaussian kernel $K(x, y; \sigma)$, where σ is the size of Gaussian kernel. The product is Gaussian-filtered image symbolized by equation (2).

$$G(x,y;\sigma) = conv2(I(x,y), K(x,y;\sigma)) \tag{2}$$

Where $conv2(\bullet)$ symbolizes the 2-D convolution procedure and Gaussian kernel is represented by equation (3).

$$K(x,y;\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

The DoG image is the difference between the two Gaussian filtered images over successive scales, as depicted by equation (4).

$$D(x,y;\sigma) = G(x,y;k\sigma) - G(x,y;\sigma) \tag{4}$$

where k is a multiplicative factor.

The convolution of image with the variable size Gaussian kernel produces a mass of blurred images with the quantity of blur dependent on scale aspect. The size of Gaussian kernel is varied using a constant multiplicative factor k.

3.2. Stable Key-point Detection

After the creation of DoG image pyramid [11], the local maxima and minima [12,15] is found by comparing every pixel with its 26 neighborhood in the 3x3 regions. These local maxima and minima feature points are shown in Fig. 7. When feature point candidates have been created, the low contrast and solid edge response points must be eliminated to make it sturdy against disturbance.

This approach gives rise to a lot of feature points depending on the image size. Quality of feature points is more important than the quantity for reliable object recognition. The features that have greater probability to be found in the other version of the image exhibit better quality. To check the stability of feature points, the local extrema are compared with a minimum threshold value. The extrema that have relatively higher minima or maxima value pass the threshold test and are considered, while the weak feature points having low contrast are discarded, resulting in less but more stable candidates.

4. SYSGEN MODEL FOR FEATURE DETECTION

The architecture developed for each of the stage feature detection and matching is first implemented in software, then feature detection based on SIFT algorithm is implemented in

SysGen model based environment [13]. This was carried on to deal with the issues concerned with that of synchronization and data type matching. Upon obtaining a simulated design, hardware Co-Simulation of the same carried on. This was intended to perform functional verification also taking into account the issues concerned with routing delays, latency and timing constraints.

Implementation of feature detection on FPGA is followed on a 'Design Module' basis [14]. Initially, the input image which is a 2-D matrix is converted into a 1-D vector. This 1-D vector from MATLAB workspace environment is passed as an input to the Gaussian filter blocks of SysGen to perform 2-D convolution operation between input image and Gaussian kernels. The add/sub blocks are used to find the difference between two Gaussian images will be taken. The results then passed through a 3x3 window generator subsystem and maxima and minima blocks. Finally using AND and OR gates strong features of the image will be detected. Then finally the resultant of the SysGen model will be a 1-D vector taken onto the MATLAB workspace. This 1-D vector is then converted back to its 2-D form using a MATLAB program to obtain the Strong Key-points. The overall SysGen model of feature detection is as shown in Fig. 2. The model shown can be categorized into the following major subsystems:

Subsystem 1: 3x3 Window generator.

Subsystem 2: Maxima and Minima Block.

Subsystem 3: 5x5 Gaussian filter.

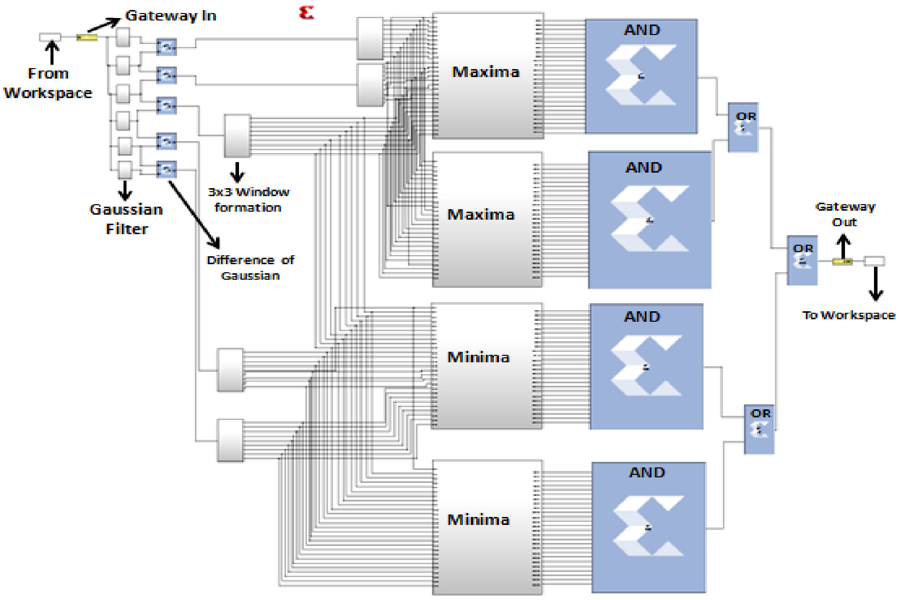


Fig. 2. SysGen model for Feature detection

4.1. Subsystem 1: SysGen Model for 3x3 window generator

This subsystem implements the 3x3 window on every DoG images. The design comprises of six delays and two virtex line buffers to form a 3x3 window on every DoG images. The virtex line buffers have a depth equal to the number of columns in an input image.The SysGen model of 3x3 window generator is as shown in the Fig. 3.

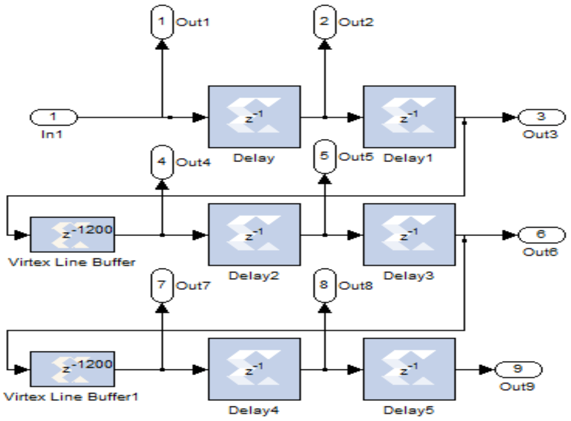


Fig. 3. SysGen model for Window generator

4.2. Subsystem 2: SysGen Model for Maxima and Minima blocks

The implementation of the maxima and minima blocks is carried out in this subsystem. These blocks are used for finding the maximum value among the 26 neighbors and employs ‘compare and select’ process. The maxima and minima are found by comparing every pixel with its 26 neighborhood in the 3x3 regions. These local maxima and minima points are treated as candidate feature points.

The design comprises of 26 Relational blocks with comparison is set to $a > b$ for maxima block and $a < b$ for minima block, to find the maximum and minimum value among the 26 neighbors. The SysGen model of maxima and minima block is as shown in the Fig. 4.

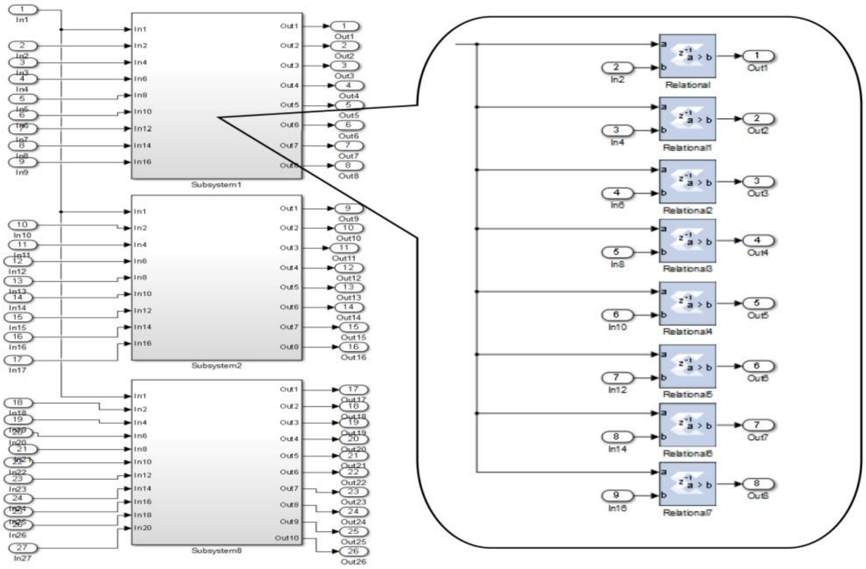


Fig. 4. SysGen model for Maxima and Minima Block

4.3. Subsystem 3: SysGen Model for 5x5 Gaussian filter

The 1-D vector from MATLAB workspace environment is passed as an input to the convolution blocks of SysGen to perform 2-D convolution operation between input image and Gaussian kernels. The resultant of the SysGen model will be a 1-D vector taken onto the MATLAB workspace. This 1-D vector is to be then converted back to its 2-D form using a MATLAB program to obtain the Gaussian filtered image. The overall SysGen model of 5x5 Gaussian filter is as shown in Fig. 5.

Convolution block subsystem gives 2-D convolution of input image with the corresponding Gaussian kernels. This design comprises of five multiplier blocks, concatenated using delay blocks and virtex line buffers with a depth equal to the number of columns in an input image. The design of a 2-D convolution block is shown in Fig. 5.

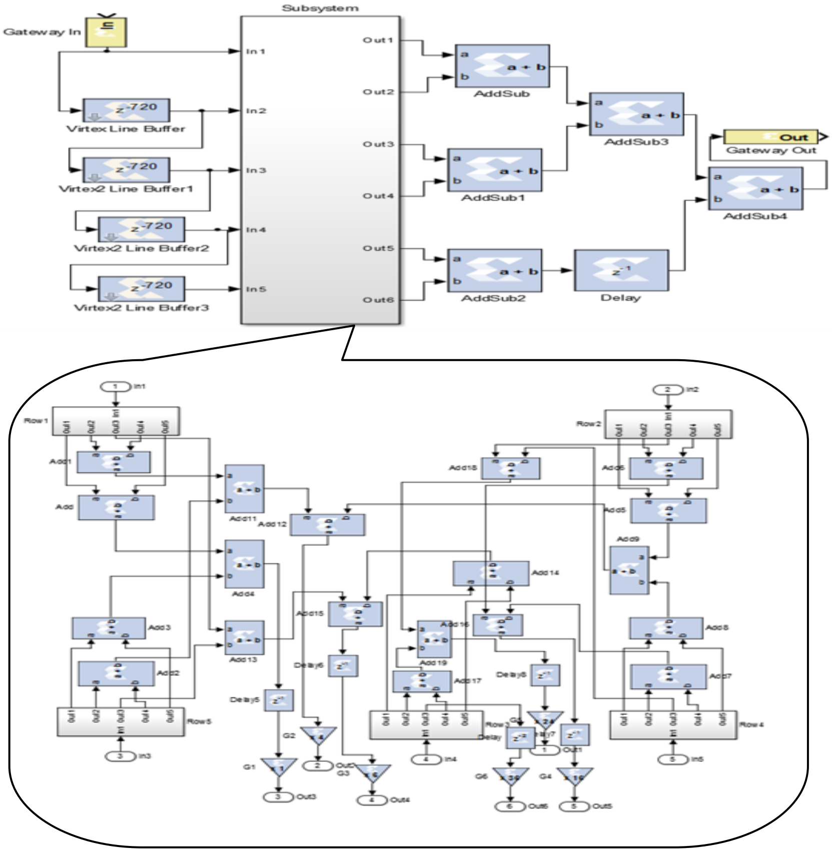


Fig. 5. SysGen model for Gaussian Filter

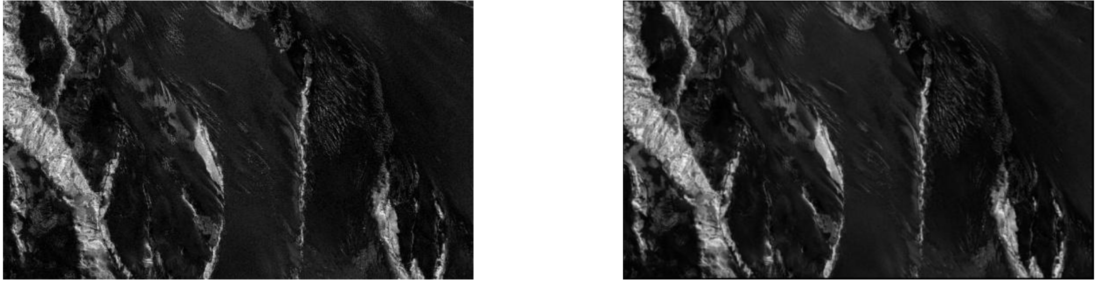
5. RESULTS AND DISCUSSIONS

The experimental result of feature detection for the above proposed architecture is presented. The performance can be considered based on three parameters: accuracy, speed and hardware utilization. There is a trade-off present between these parameters; higher accuracy consumes more hardware resources and processing time. In this work, the preference is given to accomplish high accuracy in real-time restraint.

Processing Time: The processing time is anticipated based on the number of clock cycles necessary to accomplish every task and the operational frequency for that particular module [16]. The processing time is anticipated as Number of clock cycles per task divided by operational frequency in Hertz.

SIFT Module Processing Time: The maximum operating frequency of the anticipated design is 85.390MHz. The SIFT feature extraction module takes 1280x720 clock cycles to examine the input image and identify the feature points.

Feature Detection: A high definition image is selected in the process of analysing the performance of architecture. The HD image shown in Fig. 6 is selected. At this stage the input image was convolved with the Gaussian kernel.



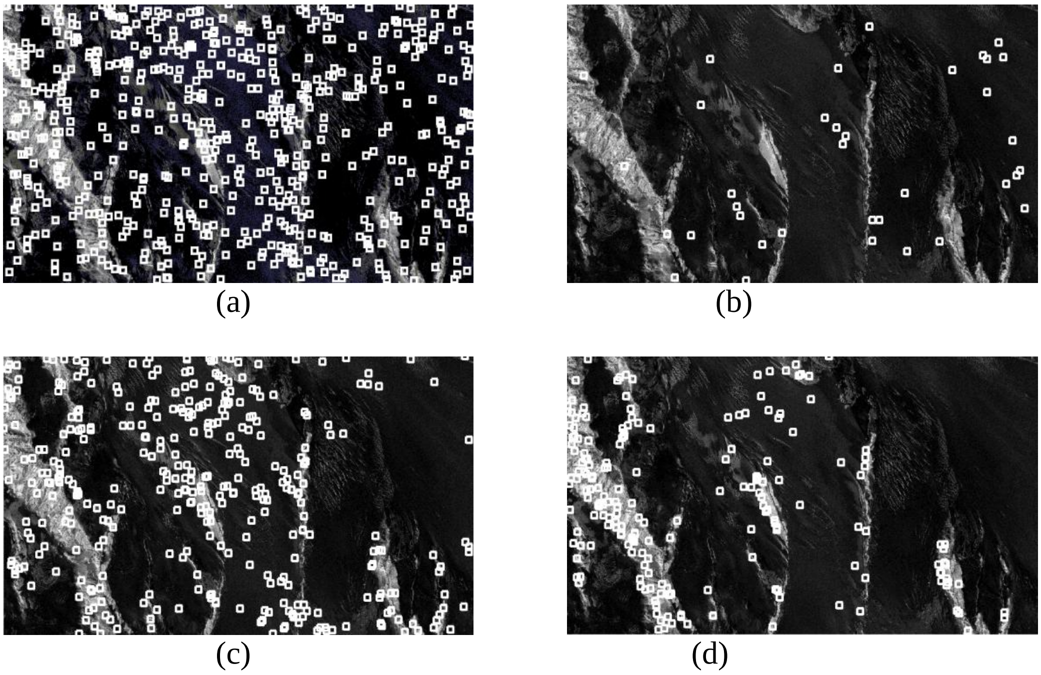
(a) (b)

Fig. 6. (a) Input image (b) Gaussian smoothened image

The next step is to take difference of Gaussian to get the minima and maxima points. It gives the co-ordinates of the pixel.

The feature points of the image are obtained. The feature points for different values of size of the Gaussian kernel or standard deviation (σ) are shown in the Fig. 7. Increase in the σ value decreases the height of the kernel while the width of the kernel increases. The above analysis is carried out for different 200 images.

The results of synthesis show that the proposed architecture requires less resources and achieves the real-time processing requirements. Speed can be further increased by using higher frequency. The system is able to achieve maximum frequency of 100MHz. Furthermore it is also possible to increase the frequency beyond 100MHz by introducing more pipeline stages at the expense of more resources. However increasing the frequency directly impacts the power requirements. In applications that require lower frame rate a lower clock can be used to get more efficiency in terms of power and resources.


(a) (b)

(c) (d)

Fig. 7. Feature points using Gaussian kernels of different variances:

(a) σ = 5 (b) σ = 5.62 (c) σ = 5.93 (d) σ = 6.25

The hardware utilized the feature detection module in the proposed architecture is given in Table 1. The FPGA essentially consists of hardware resources such as memory, slice registers, slice LUTs, LUT flip flop pairs and DSP blocks [11,12]. The results are reported from the synthesis reports generated by Xilinx ISE environment.

Table 1. Utilized Hardware for the Feature detection

Device Utilization Summary (estimated values)			
Logic Utilization	*Used*	*Available*	*Utilization*
Number of Slices	19487	25350	7.68%
Number of Slices LUT	5345	101400	5.27%
Number of Slices Registers	8427	202800	4.15%
LUT as Flip Flop pairs	6959	101400	6.86%

Number of BRAMs	34	350	10.46%
Number of IOBs	33	400	8.25%

The input image size of 1280x720 is considered with five Gaussian levels and a kernel size of 5. The effect of changing number of octaves is negligible since the resources are time shared among the octaves. The maximum operating frequency of hardware is 100 MHz and minimum period is 10ns. The total time to extract the number feature points does not depend on the number of feature points. For this design, we have achieved 108 frames per second for HD-720p video which is much above the real-time processing requirement.

This gives an overview of synthesis results for the target FPGA Xilinx Genesys 2 Kintex – 7 XC7k325t (Package: fbg676, Speed grade: -1). Xilinx Vivado (version 2014.4) has been used for the synthesis of design. Table 1 recapitulates the hardware resources used in the implementation the of the feature detection module.

CONCLUSION

The hardware architecture for feature detection based on Scale Invariant Feature Transform (SIFT) is proposed in this paper. The design of computationally effective hardware architecture for feature recognition and coordinating system on a solitary FPGA chip is projected. This system has the capability to detect SIFT features, extract the descriptors for the detected features and complete features matching for two images taken at separate standpoints, rotation, scaling and change in illumination. The real time feature identification and matching for a series of images are accomplished.

The architecture is implemented on a Xilinx Genesys 2 Kintex - 7 FPGA. The results presented shows that the hardware is reliable and supports real-time applications on an HD image size of up to 1280x720. The maximum operating frequency of hardware is 100 MHz and minimum period is 10ns. It is possible to use this feature extraction hardware in real-time high definition video applications.

REFERENCES

[1] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *Int. J. Computer Vision*, 60(2), 91-110.

[2] Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features, *Proc. of the Seventh IEEE Int. Conf. on Computer Vision*, Kerkyra, Greece, 1150-1157.

[3] Jian Wu, J. Cui, Z., Sheng, V. S., Zhao, P., Su, D. & Gong, S. (2013). A Comparative Study of SIFT and its Variants, *Measurement Science Review*, 13(3). https://doi.org/10.2478/msr-2013-0021

[4] Harris, C. (1988). A combined corner and edge detector, *Proc. 4th Alvey Vision Conf.*, Manchester, UK, 147-152.

[5] Wang, J., Zhong, S., Yan, L. & Cao, Z. (2014). An Embedded System-on-Chip Architecture for Real-time Visual Detection and Matching, *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3), 525-538.

[6] Mishra, P., Nidhi A.I., Kishore, J.K., Nandini, S. & Iffat, U. (2014) Embedded Hardware Architectures for Scale and Rotation Invariant Feature Detection, *Proc. of IEEE Int. Conf. Electronics, Computing and Communication Technologies* (IEEE CONECCT), Bangalore, India.

[7] Alhwarin, F., Wang, C., Risti-Durrant, D. & Graser, A. (2008). Improved SIFT-Features Matching for Object Recognition, *Proc. of BCS Int. Academic Conf.*, London, UK, 179-190.

[8] Wang, Z., Xiao, H., He, W., Wen, F. & Yuan, K. (2013) Real-time SIFT-based Object Recognition System, in *Proc. IEEE Int. Conf. on Mechatronics and Automation* (ICMA), Takamatsu, Japan, 1361-1366.

[9] Cheung, W. & Hamarneh, G. (2009). n-SIFT: n-Dimensional Scale Invariant Feature Transform, *IEEE Trans. on Image Processing*, 18(9), 2012-2021.

[10] Tuytelaars, T. & Mikolajczyk, K. (2007). Local Invariant Feature Detectors: A Survey, *Foundations and Trends in Computer Graphics and Vision*, 3(3), 177-280.

[11] Raut, N.P. & Gokhale, A.V. (2013). FPGA Implementation for Image Processing Algorithms Using Xilinx System Generator, *IOSR J. of VLSI and Signal Processing (IOSR-JVSP)*, 2(4), 26-36.

[12] Swaraj, D. & Madhumati, G.L. (2014). FPGA Implementation of SIFT Algorithm Using Xilinx System Generator, *Int. J. Emerging Trends in Electrical and Electronics*, 10(10), 80-85.

[13] Qasaimeh, M., Sagahyroon, A. & Shanableh, T. (2014). A Parallel hardware architecture for Scale Invariant Feature Transform (SIFT), *Int. Conf. Multimedia Computing and Systems (ICMCS)*, Marrakech, Morocco, https://doi.org/10.1109/ICMCS.2014.6911251 .

[14] Bonato, V., Marques, E. & Constantinides, G.A. (2008). A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection, *IEEE Trans. Circuits and Systems for Video Technology*, 18(12), 1703-1712.

[15] Rajesh, N., Kulkarni, R.R., Sarojadevi, H. (2014). Hardware Architecture for Scale and Rotation Invariant Feature Detection for Image Registration, *Int. Conf. Emerging Research in Computing, Information, Communications and Applications*, 239-244.

[16] Zhong, S., Wang, J., Yan, L., Kang, L., Cao, Z. (2013). A real-time embedded architecture for SIFT, *J. Systems Architecture*, 59, 16–29.