IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

Real-time FPGA-based Visual Feature Extraction using FAST and BRIEF Algorithms

A. Al-Marakeby

Systems and Computers Engineering Dept. , Faculty of Engineering , Al-Azhar Univ. Cairo, Egypt

Abstract: Visual feature extraction is widely used in many of computer vision algorithms such as object detection, stereo matching, image matching, and Visual SLAM. Real-time implementation of visual feature extraction suffers from long latency, and heavy computation. GPUs are commonly used to accelerate computationally intensive applications, but they are power hungry devices and, they have a fixed level of parallelism. To achieve processing optimization for such computationally intensive applications, FPGA, and ASIC are more efficient. In this work, an efficient and optimized FPGA architecture is designed to accelerate the computation of visual feature extraction. FAST and BRIEF algorithms are used in this system because they are simple and efficient when working with mobile and embedded systems. The system uses different frequency clocks for input pixels streaming and for processing to prevent stall cycles and achieve high speed. Multi pixels are processed per single clock cycle using optimized parallel and pipelined architecture. The proposed architecture is implemented on TERASIC DE2-115 FPGA board and tested with different image sizes and achieved very high throughput. The system can detect features up to 1000 frames per second for grayscale images with the size of 480 ×480 pixels.

Keywords: feature extraction, Parallel image processing, computer vision , FPGA, real-time processing.

I. INTRODUCTION

Feature extraction is an important part of several computer vision applications such as object tracking, object recognition, simultaneous localization, and mapping (SLAM), augmented reality, and image registration. Feature extraction (including keypoint detection and descriptor generation) is a computationally intensive process which represents a challenge when working in real-time systems. Two solutions can be used to accelerate the computation of feature extraction: 1) using faster and simpler algorithms, 2) using parallel and fast hardware platforms and architectures. There are many algorithms used in feature extraction with different characteristics and performances. Many algorithms such as scale-invariant feature transform (SIFT) , speeded up robust features (SURF) and histogram-of-gradients-based (HoG) are computationally expensive and they require huge memory access and storage[11][16][17]. The binary descriptors are more computationally efficient and requires less memory[4][13]. Also, they are simpler in matching which make them more suitable in embedded systems. Binary Robust Independent Elementary Features (BRIEF) is a simple and efficient descriptor which works fast in generation and also in matching[4][10]. In this work, features from accelerated segment test (FAST) is used to detect the keypoints, and Binary Robust Independent Elementary Features (BRIEF) is used as a descriptor. Using FAST and BRIEF algorithms, increases the speed and achieves real-time performance. The proposed feature extraction system is implemented on Field programmable Gate Array (FPGA) using customized and optimized parallel architecture. FPGAs have the advantages of low power consumption and small size making them suitable for embedded and mobile applications. In this architecture, optimized pipelined stages are used to increase the processing clock frequency and achieve high throughput. Two different clock frequencies are used : high frequency for input pixels streaming , and low frequency for processing and computation modules. FIFOs (First-In First-Out) with wide data buses are used to transfer multi pixels per each clock cycle. Parallel modules are implemented in the detection phase to process several pixels concurrently. Fixed-point approximation is used to avoid the complexity of floating-point calculations. The data required to compute BRIEF descriptor are prepared using 31x31 register bank to avoid any latencies required in memory transfers. The proposed architecture achieved very high speed (more than 1000 fps for 480x480 grayscale images. The rest of this paper is organized as follows. Section 2 gives related work. FAST and BRIEF algorithms are discussed in section3. The hardware architecture and its implementations are presented in Section 4. Section 5 presents the experimental results, and finally, a conclusion is given in Section 6.

II. RELATED WORK

Implementations of real-time feature extraction can be achieved using different hardware platforms such as : Graphics Processing Units (GPUs) , multi-core processors , Application Specific Integrated Circuits (ASICs) , and FPGAs [6]. The multi-core processors and GPUs are commonly used to speed up the computation of image processing and computer vision algorithms [7][9][12]. ASICs and FPGAs are flexible and more power efficient. Many researchers

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

have investigated the FPGA implementation of features extraction and/or matching for real-time applications [1][3][8][10]. In [1] a FPGA architecture is proposed which extracts features for Full HD images with a speed of 44 frames/s. They used non-textured corner filter combined to a subpixel refinement. Huang et.al have designed an FPGA architecture for FAST and BRIEF algorithm for on-board corner detection and matching[10]. They have achieved a processing speed of 310 frames/s for 512 x 512 image size. Liang et al. have designed a fast SIFT real-time visual feature extraction and matching[6]. The implemented FPGA-based system is capable to compute 2000 feature points for HD1080p30 at 100 MHz, using SIFT based feature extraction. Wenping et. al have designed a binary-descriptor-based image feature extraction accelerator[17]. The proposed accelerator is implemented in the Verilog hardware description language and verified on a FPGA platform. Also the accelerator is fabricated by the TSMC 65-nm CMOS technology. The accelerator achieved 135 frames/s in 1080p resolution while operating at 200 MHz.

III. FEATURE EXTRACTION

Robust and fast visual features are very important in computer vision. Generally, Feature extraction consists of two stages: feature detection and feature description as shown in fig.1. Feature detectors are used to locate points of interest in an image[2][14]. Feature descriptors encode information about point of interest into numerical values that can be used to differentiate one feature from another. Different techniques are available such as SIFT, SURF , BRIEF , and ORB (Oriented FAST and Rotated BRIEF). Some techniques are used for detections only , while some techniques are used for both of the detection and description phases. Ideally, feature descriptors should tolerate illumination changes, motion blur , pose variation, and other typical scene changes and distortions. However not all descriptors have all these features. Table.1 shows some of the commonly used techniques and their features. Although, some algorithms are robust and invariant to rotation and scaling , but they are very complex and computationally intensive. FAST and BRIEF algorithms are used in this work to achieve real time performance.

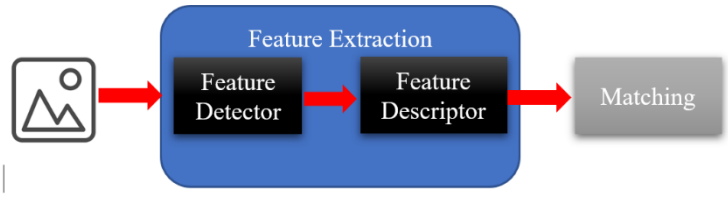


Fig.1 Feature extraction process.

TABLE 1
COMPARISON OF COMMON FEATURE EXTRACTION TECHNNIQUES

Algorithm	Function	Scale Invariant	Rotation Invariant
FAST	Detector	-	-
BRIEF	Descriptor	No	No
ORB	Detector + Descriptor	Yes	Yes
BRISK	Descriptor	Yes	Yes
LDB	Descriptor	No	No
SIFT	Detector + Descriptor	Yes	Yes
SURF	Detector + Descriptor	Yes	Yes
FREAK	Descriptor	Yes	Yes

A. FAST Detector

There are many corner detection algorithms, but due to their computational complexity few algorithms are suitable for real-time applications. The FAST(features from accelerated segment test) algorithm was introduced as a compromise between the quality and the speed of corner detection[11][15]. A pixel p is tested by examining a circle of 16 pixels (a Bresenham circle of radius 3) surrounding p [15]. Fig. 2 illustrates the FAST detector. A feature is detected at pixel p if there exists a set of n contiguous pixels in the circle, which are brighter than Ip (the intensity of this pixel) plus a threshold T, or are darker than Ip minus the threshold T [10][15].

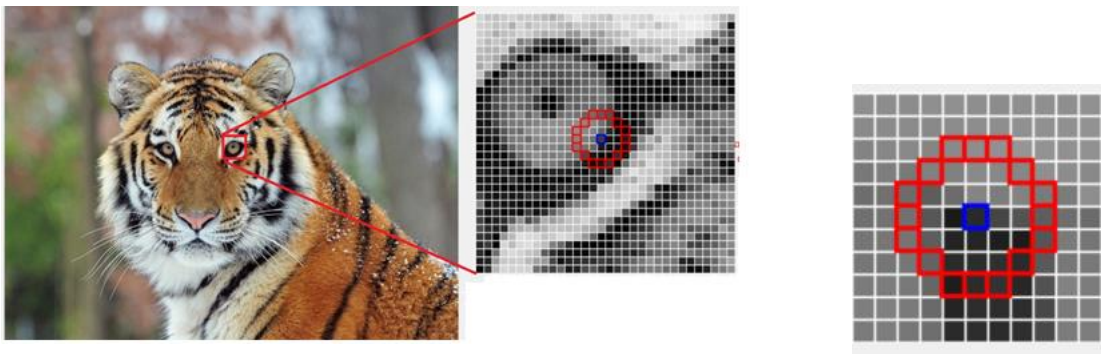
IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401



a) b)

Fig.2 FAST detector a) input image b) Bresenham circle of radius 3

The FAST detector is given by the following formula :

$$S_{p\to x} = \begin{cases} b, & I_{p\to x} \geq I_p + T \ \ (brighter) \\ s, & I_p - T < I_{p\to x} < I_p + T \ \ (similar) \\ d, & I_{p\to x} \leq I_p - T, \ \ (darker) \end{cases} \quad (1)$$

where Ip is the intensity of p, $I_{p\to x}$ is the intensity of pixel x and T is a threshold. The pixels surrounding pixel p can be classified into three groups : b (brighter) , s (similar) ,or d (darker). If there exist n continuous pixels that belong to the brighter or darker group, p is regarded as a corner[10]. The FAST algorithm calculate also a point score which is used to compare adjacent corner points to suppress unimportant corners. To calculate the score equations 2-4 are used to calculate members and scores for both bright and dark, while TH $=I_p + T$, TL $=I_p - T$, DBx=Ipx-TH, and DBx=TL -Ipx .

$$Members_B = \begin{bmatrix} I_{p1} > T_H \\ I_{p2} > T_H \\ I_{p3} > T_H \\ . \\ . \\ I_{p16} > T_H \end{bmatrix} \quad (2)$$

$$Members_D = \begin{bmatrix} I_{p1} < T_L \\ I_{p2} < T_L \\ I_{p3} < T_L \\ . \\ . \\ I_{p16} < T_L \end{bmatrix} \quad (3)$$

$$score_B = (Members_B)^T . \begin{bmatrix} DB1 \\ DB2 \\ DB3 \\ . \\ . \\ DB16 \end{bmatrix} \quad (3)$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

$$score_D = (Members_D)^T . \begin{bmatrix} DD1 \\ DD2 \\ DD3 \\ . \\ . \\ DD16 \end{bmatrix} \quad (4)$$

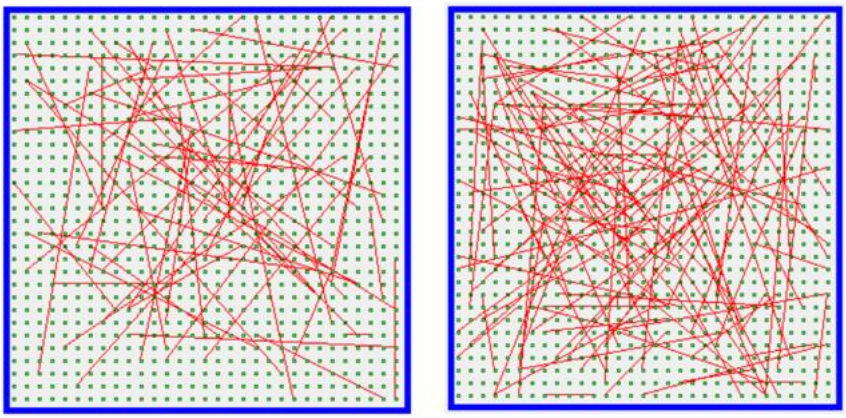
To remove adjacent corners, NMS (Non-Maximal Suppression) is used to determines if a pixel is a corner or not by comparing its score value to the score values of its adjacent pixels[11]. The point with maximum score is chosen as a corner while other adjacent points are rejected.

B. BRIEF Descriptor

The Binary Robust Independent Elementary Features(BRIEF) descriptor first proposed by M. Calonder [4] is used to describe the detected feature points. BRIEF descriptor is very fast in both of building and matching which make it suitable for real-time applications. The BRIEF descriptor consists of a bit string with a dimension nd, where nd generally equals to 64, 128 , 256 or 512 bits [10][5].
To compute BRIEF descriptor a test τ is defined on patch p of size S x S as

$$\tau(p;\ x_1, y_1, x_2, y_2) = \begin{cases} 1 & if\ I(p; x_1, y_1) < I(p; x_2, y_2) \\ 0 & if\ I(p; x_1, y_1) \geq I(p; x_2, y_2) \end{cases} \quad (5)$$

where $I(p; x_1, y_1)$ and $I(p; x_2, y_2)$ are the intensity of the pixels at (x1,y1) and (x2,y2) respectively. By choosing a set of nd location pairs uniquely defines a set of binary tests[5].



a) b)

Fig.3 different pairs sets for BRIEF descriptors, S=31, a) nd =64 b) nd =128

To reduce the sensitivity of BRIEF descriptor to noise , a smoothing kernel can be used. Pre-processing of the patch by applying Gaussian kernel, achieves better performance[5].

IV. HARDWARE ARCHITECTURE

FPGAs are flexible and fast platforms, which allow parallel and power efficient computation, making them suitable hardware platform for many image processing and computer vision applications. Fig. 4 illustrates the proposed FPGA-based architecture for real-time implementation of FAST detector and BREIF descriptor. The external memory stores the input image , and a reading module is used to transfer data from external memory to on chip buffer. A FIFO (First In First Out) consists of 7 memory based line buffers is used to transfer data from on chip buffer to 7x7 register bank. The 7x7 register bank is used as an input to the FAST detector to detect corner points. The same 7x7 register bank is used as input for the Gaussian filter. A second stage of FIFO with 31 memory-based line buffer is used to construct the patch used in BREIF descriptor.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

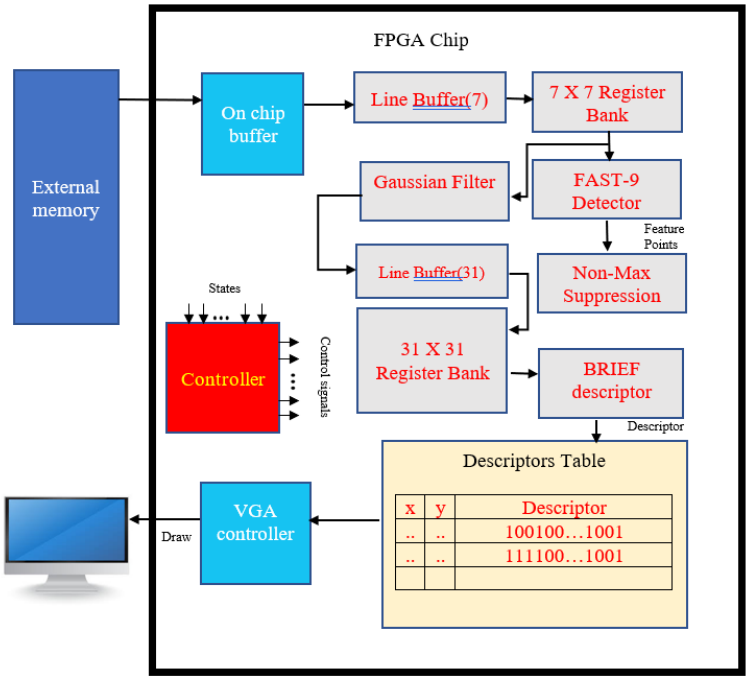


Fig.4 FPGA architecture of real-time FAST and BRIEF detector

This line buffer transfers image data to 31x31 register bank which store the patch around the corner pixel. The BRIEF descriptor module calculates the binary string based on the pairs comparison of the patch stored in 31x31 register bank. Descriptors table stores both of the detected corners locations and the BRIEF descriptor as a string of ones and zeros. The input image is displayed on an external monitor using the VGA module which also draw the locations of the corners based on the data stored on descriptors table. The controller module receives states and data from different modules and sends all required control signals. Non-Maximal Suppression module compares the score of FAST algorithm to other neighbours to minimize multiple responses to a single corner.

A. FAST detector Architecture

Fig.5 and Fig.6 illustrate the corner detection modules. The 7x7 register bank stores the pixels intensities required to calculate the members and scores in FAST algorithm. The bright and dark modules shown in fig.6 receive the 16 pixels intensities (Bresenham circle) , and the center pixel intensity then generates member and scores. If There are 9 connected members, the C9 module detects them and the maximum of the dark and bright score will be considered as the final score of this point. The final result is determined by Non-Maximal Suppression module which sends a signal to the controller indicating if this pixel is a corner. The controller sends to the BRIEF module to calculate the descriptor for this point.

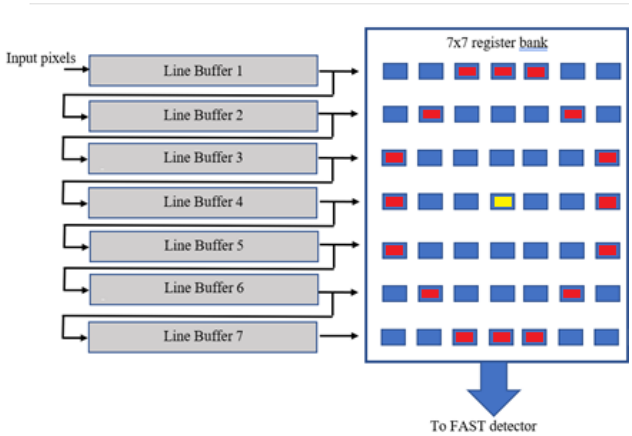


Fig.5 First Stage FIFO consists of 7 line buffers

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

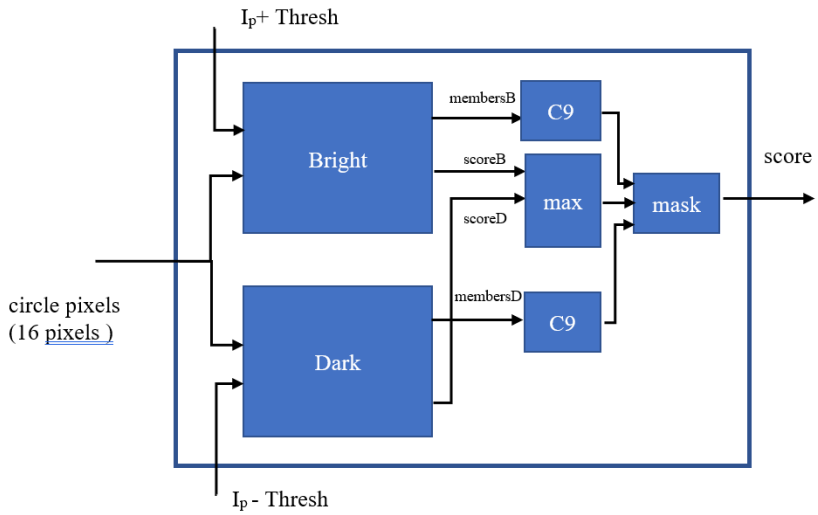


Fig.6 FAST detector module

B. BRIEF Descriptor Architecture

The Brief descriptor module consists mainly from parallel comparators which compares the pixels pairs and generates the bit string based on this comparison as shown in fig.8. The data required for BRIEF calculations is available in the 31x31 register bank shown in fig 7. To build this register bank a 31 memory-based line buffers are used. The size of each line is equal to the image width. The output string is stored in the descriptor table which is large enough for storing thousands of detected feature points.

C. Gaussian filter

The Gaussian filter decreases the sensitivity of BRIEF descriptor to noise. Gaussian filter is implemented using fixed point approximation to avoid the complexity of floating-point arithmetic. The 2D Gaussian function with a standard deviation σ can be described by

$$f(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}} \qquad (5)$$

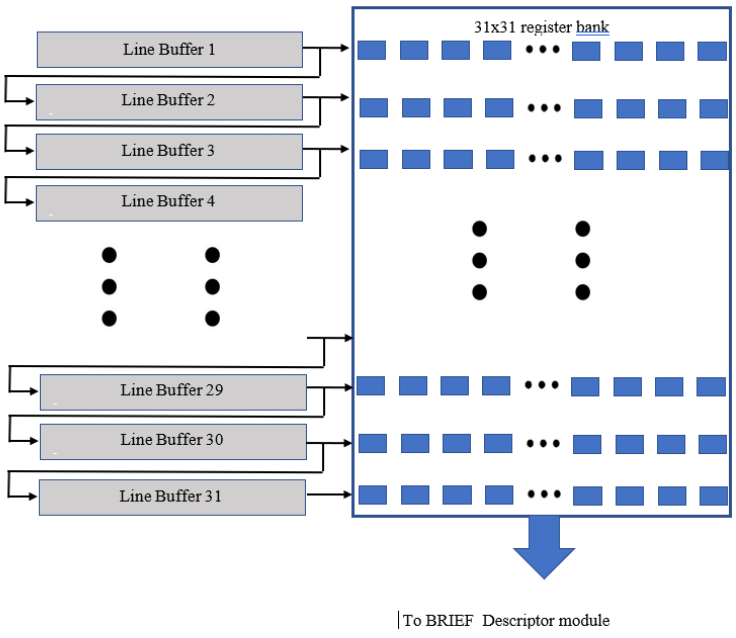


Fig7 Second Stage FIFO consists of 31 line buffers

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

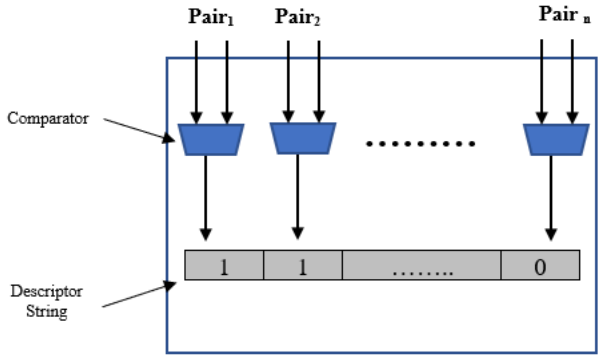


Fig.8 BRIEF Descriptor module

Different kernel sizes are commonly used in 2D Gaussian filter such as 3x3 , 5x5 , and 7x7. In this work the 7x7 kernel is used which receives the data from the same register bank used in FAST detector module. The fixed point approximation is for 2D gaussian filter is shown in fig.9.

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

Fig.9 Fixed Point approximation of 7x7 2D Gaussian filter

D. Parallel and pipelined architecture

The architecture shown in fig.4 is improved using optimized pipeline stages and parallel architecture. To increase the processing speed, the critical path found in FAST detector is divided into multiple pipeline stages. The bright and dark modules consist of 16 Adder modules which are arranged in 4 levels pipelined adder tree to increase the clock frequency. The available resources in FPGA allows the implementation of multiple FAST detector modules easily. However, The line buffer data width is 8 bit which transfer only one pixel per each clock cycle making no meaning for implementing several FAST detector modules. To solve this problem wider line buffers are used. Fig 10 illustrates the using of 64 bit line buffer which achieve a throughput of 8 pixels for each clock cycle. Fig. 11 shows the parallel FAST detection architecture which allows testing 8 pixels concurrently. The proposed architecture allows using different clock frequencies for streaming and processing. If the processing clock frequency is low due to the latency in different modules, the data can be copied from external memory using higher frequencies. In this architecture the streaming frequency can be 4 times the processing frequency. The memory data bus is 16 bit which copies 2 pixels/clock. The results of the parallel FAST detection are tested in Non-Maximal Suppression module, then the final result are send to the controller. The controller sends a signal to the BRIEF detection module to compute the descriptor if a corner is detected. The final results include the coordinates of the feature point and the descriptor string is stored in the descriptor table.

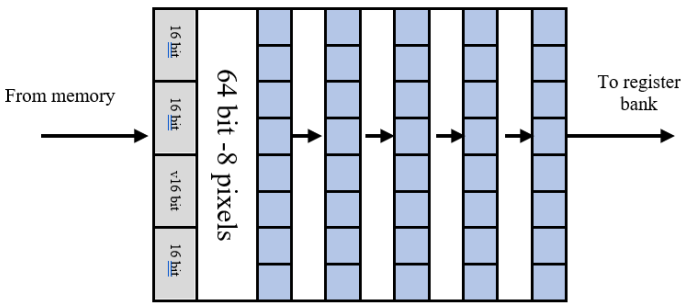


Fig. 10 64 bit - 8 pixel FIFO

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

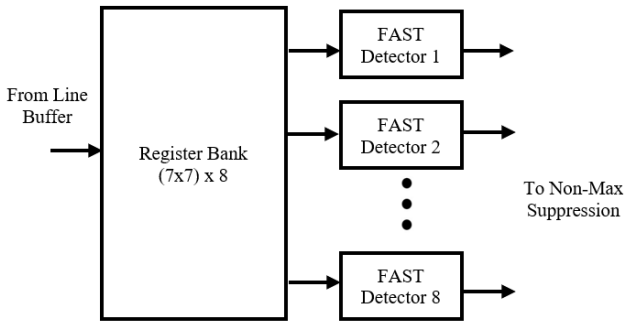


Fig.11 parallel corner detection using 8 FAST detection modules.

V. EXPERIMENTS AND RESULTS

FPGA board DE-115 is used to implement the feature extraction system . The system is implemented in Verilog. A 480 x 480 grayscale image is used to test the system. The image is stored in FLASH memory then, it is copied to the SRAM in the initialization phase. The SRAM has 16 bit data bus which transfers 2 pixels/clock. A Ram-based on chip buffer is used to store the current portion of the image. This buffer has two different read/write ports working with different clock frequencies to allow increasing the streaming frequency. Also, the read port and write port data bus widths are different . The write port data bus is 16 bit to match the external memory, while the read port is 64 bit to match the line buffer. The clock frequency of the processing is 50MHz which allows using 200 MHz for streaming. However, the limitation of the SRAM chip speed has restricted the frequency to 120MHz. Fig. 12 shows the setup of the FPGA-based feature extraction system, where the FPGA board is connected to eternal monitor. This monitor displays both of the original image stored in SRAM and the part of the image stored in on chip buffer. The detected feature points are drawn in red color for the image part stored in the on chip buffer. All processing and computations run with the full streaming speed without any stalling. The achieved throughput is 1,076 frames/s for 480 x480 grayscale image size. This is due to the using of fast and parallel architecture which process 8 pixels/clock. This throughput can be increased easily using faster or wider data bus external memory. The processing speed is fast enough to process 1,700 frames per second, but the memory represents the bottleneck in this board. Also, the available logic and memory resources in the FPGA chip can be used to increase the parallel modules, from 8 parallel modules to 16 or more. In these cases, the throughput can be increased more and more.

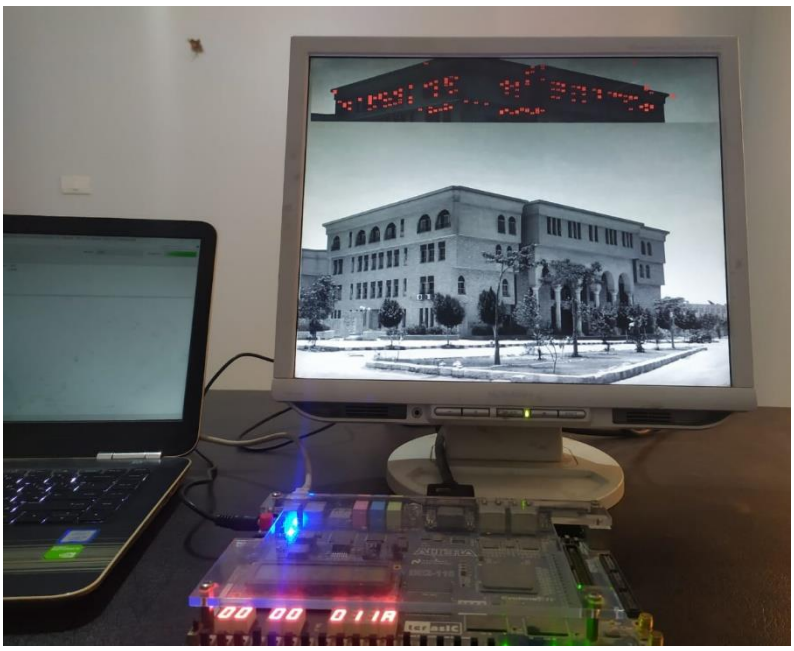


Fig.12 FPGA-based implementation of real time feature extraction

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

TABLE 2
THROUGHPUT AND LATENCY FOR DIFFERENT IMAGE SIZES

Image size	Image type	Throughput (fps)	Latency (ms)
640x480	Grayscale	781	5
480x480	Grayscale	1,076	3
320x320	Grayscale	2,343	2
320x240	Grayscale	3,125	1.5

Table 2 illustrates the throughput and latency for different grayscale image sizes. The results show the high throughput of the proposed architecture which achieve more than 1000 fps for 480x480 image size. In [11] , the feature detection only without feature description works with 60 frames per second for 1920 x 1080 HD. In table 3, the FPGA resource utilization is illustrated using ALTERA EP4CE115F29C7 CHIP and using 480 x 480 grayscale images. The memory bits utilization depends on the image size where the FIFO size is related to the image width.

TABLE 3
RESOURSE UTILIATION USING ALTERA EP4CE115F29C7 CHIP AND USIGN 480X480 IMAGE SIZE

Resources	Utilization	Ratio
Logic Elements	5,322/114,480	5%
Registers	5507	-
Memory bits	964,472/3,981,312	24%
Embedded multipliers 9-bit	0/512	0%

For high resolution images, it is expected to use more memory bits. It is clear that the logic elements utilization is very low, which allows increasing the parallelism by adding more parallel modules.

VI. CONCLUSION

In this work, we have presented a fast parallel architecture for visual feature extraction using FAST and BRIEF algorithms. The proposed architecture achieved high throughput with thousands of frames per second. Using two different cock frequencies for both of streaming and processing prevents stall cycles and increase the processing speed. Using wide FIFOs with 64 bit word lengths allowed the processing of multiple pixels/ clock. The fixed-point calculations have simplified of Gaussian filter module without significant accuracy losses. Also, higher throughput can be obtained using faster streaming. The proposed system reduces the latencies found in real-time feature extraction systems and allow the implementation of fast, power efficient and portable embedded systems.

REFERENCES

[1] Aguilar-González, Abiel, Miguel Arias-Estrada, and François Berry. "Robust feature extraction algorithm suitable for real-time embedded applications." Journal of Real-Time Image Processing 14.3 (2018): 647-665.
[2] Azimi, Ehsan, et al. "A fully pipelined and parallel hardware architecture for real-time BRISK salient point extraction." Journal of Real-Time Image Processing 16.5 (2019): 1859-1879.
[3] Bonato, Vanderlei, Eduardo Marques, and George A. Constantinides. "A parallel hardware architecture for scale and rotation invariant feature detection." IEEE transactions on circuits and systems for video technology 18.12 (2008): 1703-1712.
[4] Calonder, Michael, et al. "Brief: Binary robust independent elementary features." European conference on computer vision. Springer, Berlin, Heidelberg, 2010
[5] Calonder, Michael, et al. "BRIEF: Computing a local binary descriptor very fast." IEEE transactions on pattern analysis and machine intelligence 34.7 (2011): 1281-1298.
[6] Chiu, Liang-Chi, et al. "Fast SIFT design for real-time visual feature extraction." IEEE Transactions on Image Processing 22.8 (2013): 3158-3167.
[7] Elmoogy, Ahmed M., et al. "SurfCNN: A Descriptor Accelerated Convolutional Neural Network for Image-Based Indoor Localization." IEEE Access 8 (2020): 59750-59759.
[8] Fang, Weikang, et al. "FPGA-based ORB feature extraction for real-time visual SLAM." 2017 International Conference on Field Programmable Technology (ICFPT). IEEE, 2017.
[9] Garbo, Alessandro, and Stefano Quer. "A Fast MPEG's CDVS Implementation for GPU Featured in Mobile Devices." IEEE Access 6 (2018): 52027-52046
[10] Huang, Jingjin, et al. "A new FPGA architecture of fast and BRIEF algorithm for on-board corner detection and matching." Sensors 18.4 (2018): 1014.
[11] Lam, Siew-Kei, et al. "Data-path unrolling with logic folding for area-time-efficient FPGA-based FAST corner detector." Journal of Real-Time Image Processing 16.6 (2019): 2147-2158

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 4, April 2021

DOI 10.17148/IJARCCE.2021.10401

[12] Li, Jincheng, et al. "Realization of CUDA-based real-time multi-camera visual SLAM in embedded systems." Journal of Real-Time Image Processing 17.3 (2020): 713-727.
[13] Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system." IEEE transactions on robotics 31.5 (2015): 1147-1163
[14] Nikolic, Janosch, et al. "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM." 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014
[15] Rosten, Edward, and Tom Drummond. "Fusing points and lines for high performance tracking." Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Vol. 2. IEEE, 2005.
[16] Šoberl, Domen, et al. "Hardware implementation of FAST algorithm for mobile applications." Journal of Signal Processing Systems 79.3 (2015): 247-256.
[17] Zhu, Wenping, et al. "A 135-frames/s 1080p 87.5-mw binary-descriptor-based image feature extraction accelerator." IEEE Transactions on Circuits and Systems for Video Technology 26.8 (2015): 1532-1543.