Accelerating the Construction of BRIEF Descriptors Using an FPGA-based Architecture

Roberto de Lima, Jose Martinez-Carranza, Alicia Morales-Reyes, Rene Cumplido
Instituto Nacional de Astrofisica, Optica y Electronica
Computer Science Department
Tonantzintla, Puebla 72840, Mexico
Email: [delima87,carranza,a.morales,rcumplido]@inaoep.mx

***Abstract*—BRIEF emerged as a novel alternative to conventional floating-point-based descriptors such as SIFT or SURF. In contrast to these descriptors, BRIEF is a descriptor represented by a binary number offering two main advantages: low memory footprint and fast descriptor comparison. These qualities make it a suitable descriptor to be implemented on a hardware architecture, where the comparison operation can be implemented efficiently via a parallel scheme. However, the construction of BRIEF involves a sequential operation in the form of a set of pairwise tests on the image intensities, and as consequence, sequential memory access is necessary. In this paper, we propose a novel way to construct the BRIEF descriptor by arranging the pairwise tests such that data retrieval from memory is exploited, thus accelerating the descriptor construction up to 4 times when compared to the sequential way.**

I. Introduction

Over the last years, local feature point descriptors proved to be efficient for several applications such as: object recognition, 3D reconstruction, image retrieval, and camera localization. Several of these computer vision tasks are now at the core of many real-time systems, as a consequence, there is a growing need to accelerate these tasks in order to achieve real-time performance. FPGA-based architectures have emerged as a way of bridging the gap in between software implementations and dedicated hardware. Thus, vision tasks can be broken down in their essential components such that some of these can be implemented on a hardware version that can be run on an FPGA.

Visual descriptors are one of those essential components in computer vision tasks, however, substantive processing time has to be invested in their construction. For instance, one of the most popular descriptor is the SIFT (Scale-Invariant Feature Transform) descriptor [1], albeit not very suitable for real-time performance because of the complex calculations that have to be carried out in order to construct this descriptor. Alternatives to SIFT have been proposed such as SURF (Speeded Up Robust Features) [2] and HOG (Histogram of Oriented Gradients) [22], among others, aiming at reducing processing time, whilst maintaining similar discrimination capabilities. Real-time performance has been achieved by means of FPGA-based hardware architecture such as those reported in [15], [16], [17], [18], [19] for SIFT and those in [11], [12], [13], [14] for SURF.

The BRIEF (Binary Robust Independent Elementary Features) descriptor [3] emerged as an alternative to histogram-based or floating-point-based descriptors such as SIFT or SURF. In contrast to these descriptors, BRIEF is a binary number constructed through a less complex process, this is, out of the response of a set of pairwise tests applied to the image intensities; with the caveat that this descriptor losses some discrimination power, the main advantages are two fold: (i) low memory footprint; (ii) fast comparison operation, which can be efficiently implemented on dedicated hardware-level processor instructions.

Despite the fact that comparison operations for BRIEF are faster than those for floating-point-based descriptors, the pairwise tests used in BRIEF involve sequential memory access, hence its construction is sequential. Thus, motivated by the need to reduce processing time, in this paper we propose a novel way to construct BRIEF descriptors by exploiting data memory organization. For the latter, the key idea is to arrange pairwise tests following a format that enables us to take advantage of memory accessing scheme. Under this approach, pairwise tests can be carried out 4 times faster than in the usual sequential way of the conventional BRIEF descriptor.

In order to describe the proposed approach, the rest of the paper is organised as follows. In the next section, related work is presented. Section III describes BRIEF algorithm and a learning method to construct a discriminant descriptor. Both, data memory organization and memory access scheme are described in Section IV. Section V describes the proposed parallel FPGA architecture. An analysis of experimental results is drawn in Section VI. Finally, conclusions and future work are presented in Section VII.

II. Related work

BRIEF requires far less storage capacity and offers much faster matching speed than conventional floating-point descriptors such as SIFT and SURF. However, these latter proved to be robust against visual transformations such as scaling and rotation transformations. BRIEF is not robust against these transformation and therefore, some approaches were proposed in order to add robustness to BRIEF.

From the above, Ethan Rublee et al. proposed a BRIEF descriptor that is invariant to rotation called ORB (Oriented FAST and Rotated BRIEF) [4], the main contributions in this work lies in adding an orientation component to the FAST [6] feature detector and proposing a learning method for choosing pairwise tests with good discrimination power and low correlation response among them.

Similar to ORB, in [20] Leutenegger et al. proposed a binary descriptor invariant to rotation and scale. It uses the AGAST corner detector [21], which is an improvement of FAST [6]. This binary descriptor is constructed by pixel comparisons whose distribution forms a concentric circle surrounding the feature. More recently, in [23], a binary descriptor based on the human retina was proposed, it received the name of FREAK (Fast Retina Keypoint).

Hardware implementation of binary descriptors have been proposed in [10], [24], [25], [26]. However, in contrast to our work, these approaches implement a sequential scheme for the descriptor construction.

III. BRIEF DESCRIPTOR

BRIEF is a feature descriptor that uses binary tests between pixels in a smoothed image patch. More specifically, if p is a smoothed image patch, corresponding binary test τ is defined by:

$$\tau(p; x, y) := \begin{cases} 1 & if\ p(x) < p(y) \\ 0 & otherwise \end{cases} \tag{1}$$

where $p(x)$ is the intensity of p at a point x. The feature is defined as a vector of n binary tests:

$$f_{nd}(p) := \sum_{1<i\leq 1} 2^{i-1}\tau(p; x_i, y_i) \tag{2}$$

In order to construct a BRIEF descriptor that presents good performance in terms of speed, storage, efficiency, and recognition rate, it is important to take into account two elements: descriptor's length and binary tests distribution. Descriptors of 128,256, and 512 bits proved to be efficient. For this reason, the proposed architecture is able to obtain BRIEF descriptor vectors of up to 256 bits. On the other hand, many different types of distributions were considered in [3] for selecting nd test locations. Figure 1 taken from [3] shows the explored distributions, where experimental results reported that G III gives the most discriminant BRIEF descriptor.

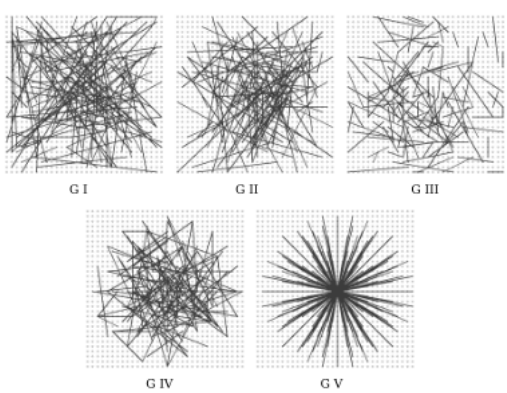

Fig. 1. Several explored binary tests distributions (figure taken from [3].

A. Learning good binary features

Rublee et al. [4], pointed out two properties presented in BRIEF, discriminative and uncorrelated binary tests. These properties are assessed with two statistic measures, mean and covariance respectively. A mean near 0.5 per descriptor's bit, gives the maximum sample variance, and as a consequence discriminative descriptors. On the other hand, a minimum covariance between BRIEF vector indicates uncorrelated binary tests. Hence, in [3], a typical Gaussian BRIEF pattern (see Figure 1,G III) was reported as the best binary test distribution.

Taking the above into account, a learning method for choosing a good set of binary tests was developed in [4]. Generically, the algorithm consists of extracting m keypoints from a set of images using FAST [6] or its variants [7]. Then, a binary test is computed for all possible pixel combinations. The resulting vector is ordered by their distance from a mean of 0.5, from which a greedy search is done with the purpose of selecting n uncorrelated tests. As a result, a BRIEF descriptor that accomplishes the desired properties is obtained. In order to validate the proposed architecture algorithmically, a binary test distribution is chosen using this method. In the next section, the proposed memory scheme is drawn in detail.

IV. MEMORY SCHEME

A. Image data allocation

As we mentioned, BRIEF algorithm implies FAST points detection and binary tests between pixels in a smoothed image patch. Therefore, it is assumed that FAST points were previously computed and stored in a single port RAM, and an 8-bit smoothed image is stored in a dual port RAM.

On FPGAs' RAM blocks, memory data is allocated in 32-bit words. Accordingly, a smoothed gray scale image is allocated in memory by clustering 4 pixels per memory address, in such a way that 8 image pixels are retrieved per clock cycle. Figure 2 clarifies the above mentioned.

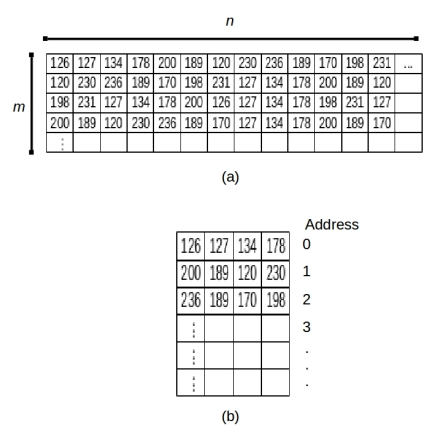

Fig. 2. a) Image with m x n resolution, b) Image memory organization

In order to obtain the memory address (l) of a specific 32-bit word. It is necessary to know the pixel's location within an image, which is given by its corresponding (x, y) coordinates, for an image with resolution of $m \times n$. This relation is described

by the following equation:

$$l = ((x-1)(\frac{n}{4})) + \frac{y-1}{4} \quad (3)$$

On the other hand, corner key points found by FAST method are allocated in a single port RAM. The first 16 memory word bits refers to the row address and the remainder to the column address. Thus, dual port and single port memory depth relies on the image size and the number of FAST points detected, respectively.

B. Memory access

Considering this scheme of image data allocation. A fast and relatively simple scheme is proposed to help avoiding sequential memory access to compute binary tests per image patch. The proposed memory accessing scheme exploits image data allocation in order to construct a BRIEF descriptor keeping both qualities explored in [4]: discriminative and uncorrelated binary tests.

The proposed scheme, consists of selecting 88 image pixels grouped in 22 memory words located within a patch around a key point value, such as in Figure 3. Image pixels location, and therefore their corresponding memory allocation, is delimited by a 24×24 image patch. Size of the image patch area is determined by the fact that pixels near key points offer relevant information. A 3 pixels offset is considered for dealing with image patch boundaries.

From selected memory words, a set of binary tests is determined based on the method described in Section II, with a set of $2k$ FAST points extracted from images of the ZuBuD set [5]. Figure 4 shows a selection example of 32 out of 256 binary tests obtained according to the proposed learning method reported in [4]. In this graphical example, random positions within 4 pixels blocks and among 22 of them are chosen in order to provide information diversity when building the binary descriptor.

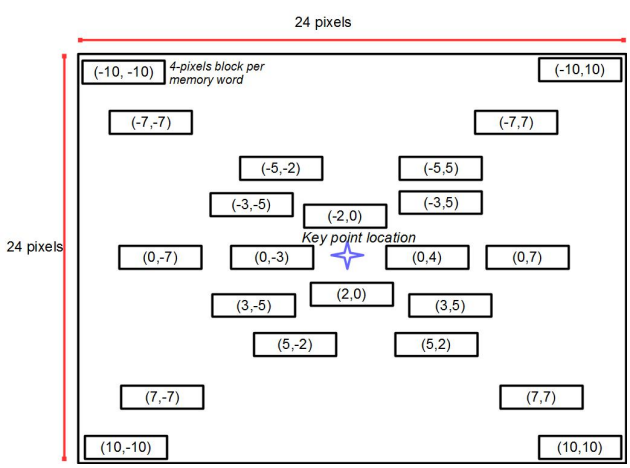

Fig. 3. Image memory words location for each keypoint.

In order to assess the binary test quality, the mean of every feature bit is calculated for all $2k$ samples. This measure is shown in Figure 5, where the blue line points out that the mean of each k-bit descriptor is close to 0.5, this numerical metric indicates that the obtained binary test vectors are good enough for discrimination.

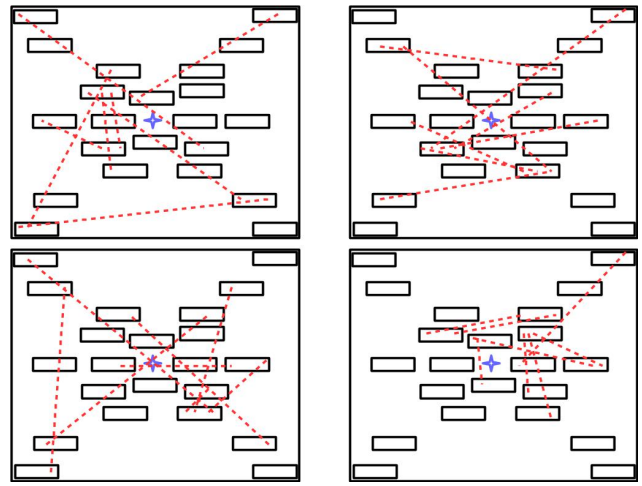

Fig. 4. Example of 32 binary tests via the learning method proposed in [4].

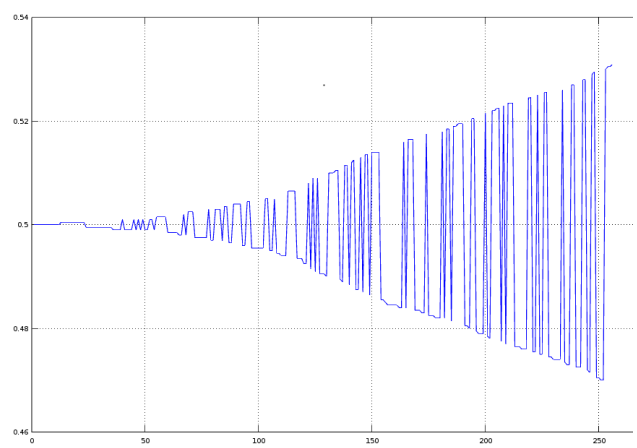

Fig. 5. Bit feature mean over 2k samples.

The proposed memory access scheme helps to construct a binary test distribution suitable to compute a quality BRIEF descriptor for discrimination while taking advantage of the retrieved image data. In the next section, the proposed hardware architecture for acceleration computation of BRIEF descriptors is described in detail.

V. Proposed Architecture

Previously, it has been detailed the proposed memory access scheme that significantly reduce the number of memory accesses needed for selecting a good set of binary tests. Motivated by the latter, we propose a parallel FPGA architecture for computing BRIEF descriptors with length of up to 256 bits. The proposed architecture has been developed using Xilinx's System Generator for Simulink and Vivado version 2014.2 and Matlab version 2014a.

An overview of the proposed parallel architecture is drawn by a block diagram in Figure 6, where FAST points and a smoothed image are stored in a single port and a dual port RAM memory, respectively. The following steps describe the process for calculating BRIEF descriptors:

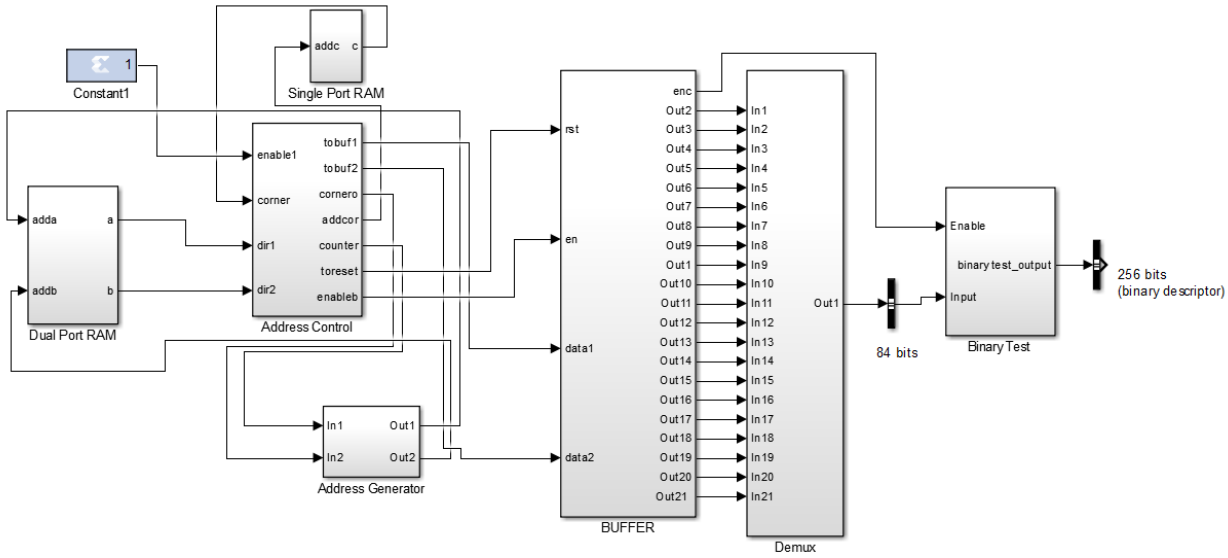

Fig. 6. BRIEF descriptor FPGA design: Overview.

1) The first corner location is read by the address control.
2) The memory address of a 4-pixels block (see Figure 3) is computed by the address generator block.
3) 4-pixels blocks are stored in a buffer.
4) Once the buffer is filled, binary tests are computed in parallel.

In the next subsections, functional description of every module presented in Figure 6 is provided.

A. Address control block

Address control block aims at synchronizing the entire process, as well as loading all 4-pixels blocks in the buffer. The design of a Mealy state machine is shown in Figure 7.

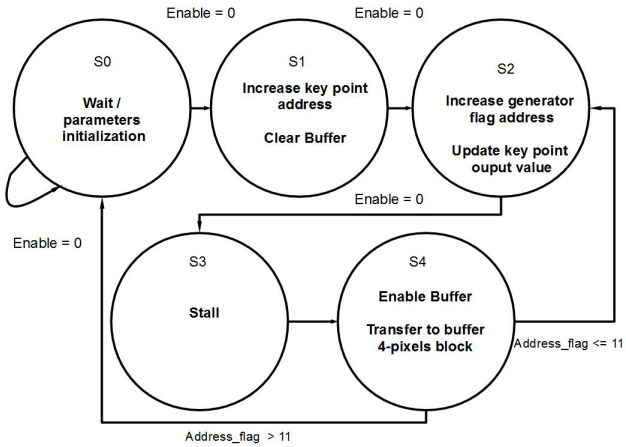

Fig. 7. Mealy state machine designed to model address control behaviour.

A Mealy state machine consist of five states. First, an address generator flag and a corresponding key point memory address are initialized in $S0$. After, a corresponding key point is chosen by increasing the corner address counter in $S1$. In $S2$, an address generator flag is increased by one unit, in order to load 22 4-pixels image blocks corresponding to current key point in the buffer. The dual port RAM memory is synchronized with the address control block by a stall. Finally, in $S4$, two 4-pixels image blocks are sent to the buffer. $S2$,$S3$ and $S4$, can be seen as a loop iteration that stops after 11 clock cycles when 22 4-pixels image blocks have been sent to the buffer. After that, the state machine returns to $S0$ and the process is repeated for the next key points.

B. Address generator block

In Section IV, it has been demonstrated that only 22 memory locations, corresponding to 4-pixels image blocks are needed for choosing a discriminative and uncorrelated set of 256 binary tests. In this way, the memory address for each key point location is computed by the address generator block. Since a key point location is known, memory addresses showed in Figure 3 are computed by adding predetermined constants to the memory address of a FAST point. It is important to mention that these constant values depend on an address generator's flag.

In order to reduce the amount of hardware resources needed to compute the address of a 4-pixels image block, only integer operations are implemented (additions and multiplications), while divisions are implemented by a 2-bit shifter.

C. Buffer and demux

A shift register is implemented with the purpose of temporarily store all 4-pixels image blocks corresponding to each FAST point before the binary test is carried out in parallel. Thus, the buffer is designed using a cascade of 21 D-flip-flops with an enable input (see Figure 8). Since 4 pixels are clustered in one memory word, a demultiplexer is used to split each 32-bit word in 4 pixels of 8 bits, see Figure 9. In consequence, once the buffer is filled, 21 demultiplexers are placed to split 22 4-pixels blocks in 88 pixels.

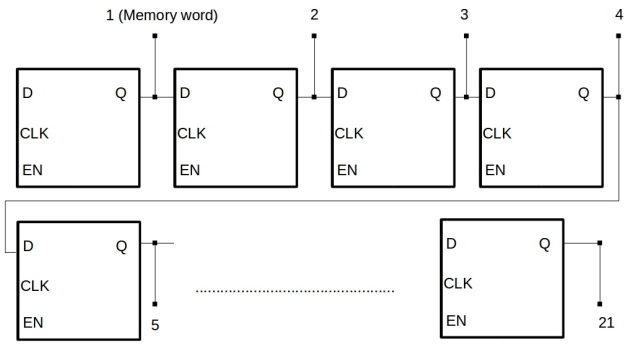

Fig. 8. Temporary storage via a shift register.

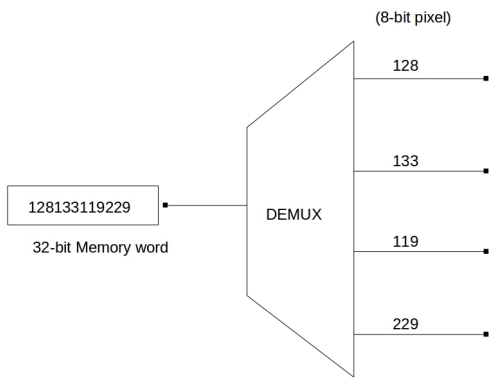

Fig. 9. 32-bit to 8-bit word demultiplexer.

D. Binary test block

Finally, binary tests are computed in parallel by implementing 256 comparators. Corresponding inputs are previously defined based on the binary tests distribution described in Section IV. As a final step, the binary test result can be stored in a RAM memory block to be used for data post-processing such as feature matching.

VI. Results

The architecture is synthesized for Xilinx Zynq XC7Z020 SoC platform. A 320×240 smoothed image is stored in dual port RAM, and 50 FAST points are stored in a single port RAM. Figure 6 gives an overview of the FPGA architecture which is assessed in terms of hardware resources and throughput. Table I shows the number of hardware resources used to compute a 256-bit BRIEF descriptor. A minimum programmable logic area is required for the whole architecture, resulting in a compact module. Calculating BRIEF descriptors is a sub-task common to complex computer vision operations. The proposed architecture is a self-contained module that can be instantiated within a more complex vision machine system.

On the other hand, as opposed to [10], a sequential memory access is avoided. A 256-bit BRIEF descriptor is calculated in 15 clock cycles, thus accelerating the descriptor construction up to 4 times. Therefore, since clock frequency is 125 MHz, for the synthesized device, the proposed hardware architecture is capable of computing BRIEF descriptors of 50 key points in 6 ms.

The most important highlights from previous results are

Resource	Utilization	Available	Utilization %
Slice LUTs	478	17600	2.72
Slice Registers	397	35200	1.13
Memory	19	60	31.67
DSP	1	80	1.25

TABLE I. Hardware Resources .

twofold: i) Since the proposed architecture requires a reduced amount of hardware resources, it can be integrated within other cores such as: feature detector, feature matching or preprocessing image stage, ii) Tests distribution chosen for FPGA architecture design is based on a data memory organization commonly used, that helps to notably reduce the rate for computing a 256-bit BRIEF descriptor, while preserving descriptor quality. From this analysis, our work is highly motivated by the scenario where an image is preprocessed and FAST points previously computed by a computer processing unit using OpenCV[9] or by a module designed in Vivado HLS using Xilinx video libraries [8]. As a final note, a comparative table is avoided, due to hardware architectures presented in related work reported results of the whole architecture, and we are only focusing on accelerating the construction of BRIEF descriptors.

VII. Conclusion

The proposed hardware architecture is designed to calculate 256-bit BRIEF descriptors based on a predefined pattern of binary tests. This approach contributes to reduce the number of memory accesses required to obtain the descriptor while maintaining its discrimination quality. This latter is assessed with the same methodology proposed by Rublee et. al. in [4]. Moreover, by following the same scheme, this architecture can be scaled to compute 512-bit descriptors by increasing the number of 4-pixels image blocks around the key point. Future work includes the FPGA implementation of FAST feature detector invariant to orientation together with a feature matching stage, and their adjustment with the current proposed architecture. It is also considered increasing flexibility through the address generator module, by enabling calculation of pixels' blocks distributions that ensure discriminant descriptors within key point patches, without requiring predefined locations.

Acknowledgments

Roberto de Lima is supported by the Mexican National Council for Science and Technology (CONACyT) studentship number 627047.

References

[1] Lowe, D. G. (2004). *Distinctive image features from scale-invariant keypoints*. International journal of computer vision, 60(2), 91-110.

[2] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *Surf: Speeded up robust features*. In Computer visionECCV 2006 (pp. 404-417). Springer Berlin Heidelberg.

[3] Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). *Brief: Binary robust independent elementary features*. Computer VisionECCV 2010, 778-792.

[4] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). *ORB: an efficient alternative to SIFT or SURF*. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2564-2571). IEEE.

[5] Shao, H., Svoboda, T., & Van Gool, L. (2003). *Zubud-zurich buildings database for image based recognition*. Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep, 260.

[6] Rosten, E., & Drummond, T. (2006). *Machine learning for high-speed corner detection*. In Computer VisionECCV 2006 (pp. 430-443). Springer Berlin Heidelberg.

[7] Rosten, E., Porter, R., & Drummond, T. (2010). *Faster and better: A machine learning approach to corner detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(1), 105-119.

[8] HLS Video Library. (n.d.). Retrieved July 30, 2015, from http://www.wiki.xilinx.com/HLS Video Library

[9] OpenCV — OpenCV. (n.d.). Retrieved July 30, 2015, from http://opencv.org/

[10] Lee, K. Y., & Byun, K. J. (2013). *A hardware design of optimized orb algorithm with reduced hardware cost*. Adv. Sci. Technol. Lett, 43, 58-62.

[11] Bouris, D., Nikitakis, A., & Papaefstathiou, I. (2010, May). *Fast and efficient FPGA-based feature detection employing the SURF algorithm*. In Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on (pp. 3-10). IEEE.

[12] Bouris, D., Nikitakis, A., & Papaefstathiou, I. (2010, May). *Fast and efficient FPGA-based feature detection employing the SURF algorithm*. In Field-Programmable Custom Computing Machines (FCCM), 2010 18th IEEE Annual International Symposium on (pp. 3-10). IEEE.

[13] Zhao, J., Zhu, S., & Huang, X. (2013, September). *Real-time traffic sign detection using SURF features on FPGA*. In High Performance Extreme Computing Conference (HPEC), 2013 IEEE (pp. 1-6). IEEE.

[14] Wilson, C., Zicari, P., Craciun, S., Gauvin, P., Carlisle, E., George, A., & Lam, H. (2014, December). *A power-efficient real-time architecture for SURF feature extraction.* In ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on (pp. 1-8). IEEE.

[15] Huang, F. C., Huang, S. Y., Ker, J. W., & Chen, Y. C. (2012). *High-performance SIFT hardware accelerator for real-time image feature extraction*. Circuits and Systems for Video Technology, IEEE Transactions on, 22(3), 340-351.

[16] Chang, L., & Hernandez-Palancar, J. (2009). *A hardware architecture for SIFT candidate keypoints detection*. In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (pp. 95-102). Springer Berlin Heidelberg.

[17] Chang, L., Hernandez-Palancar, J., Sucar, L. E., & Arias-Estrada, M. (2013). *FPGA-based detection of SIFT interest keypoints*. Machine vision and applications, 24(2), 371-392.

[18] Shao, A. J., Qian, W. X., Gu, G. H., & Lu, K. L. (2015, August). *Real-time implementation of SIFT feature extraction algorithms in FPGA*. In International Conference on Optical Instruments and Technology 2015 (pp. 96220V-96220V). International Society for Optics and Photonics.

[19] Jiang, J., Li, X., & Zhang, G. (2014). *SIFT hardware implementation for real-time image feature extraction*. Circuits and Systems for Video Technology, IEEE Transactions on, 24(7), 1209-1220.

[20] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). *BRISK: Binary robust invariant scalable keypoints*. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2548-2555). IEEE.

[21] Mair, E., Hager, G. D., Burschka, D., Suppa, M., & Hirzinger, G. (2010). *Adaptive and generic corner detection based on the accelerated segment test*. In Computer VisionECCV 2010 (pp. 183-196). Springer Berlin Heidelberg.

[22] Dalal, N., & Triggs, B. (2005, June). *Histograms of oriented gradients for human detection*. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[23] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). *Freak: Fast retina keypoint*. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 510-517).

[24] Viswanath, P., Swami, P., Desappan, K., Jain, A., & Pathayapurakkal, A. (2014, November). *ORB in 5 ms: An Efficient SIMD Friendly Implementation*. In Computer Vision-ACCV 2014 Workshops (pp. 675-686). Springer International Publishing.

[25] Lee, K. Y. (2014). *A Design of an Optimized ORB Accelerator for Real-Time Feature Detection*. International Journal of Control & Automation, 7(3).

[26] Bello, E. D., & Salvadeo, P. (2014, November). *An image descriptors extraction hardware-architecture inspired on Human Retina*. In Programmable Logic (SPL), 2014 IX Southern Conference on (pp. 1-6). IEEE.