Machine Vision and Applications
DOI 10.1007/s00138-017-0851-5

ORIGINAL PAPER

Improving the construction of ORB through FPGA-based acceleration

Roberto de Lima[1] · Jose Martinez-Carranza[1] · Alicia Morales-Reyes[1] · Rene Cumplido[1]

Received: 4 May 2016 / Revised: 10 May 2017 / Accepted: 25 May 2017

Abstract Binary descriptors have won their place as efficient and effective visual descriptors in several vision tasks. In this context, one of the most widely used binary descriptors to date is the ORB descriptor. ORB is robust against rotation changes, and it uses a learning procedure to generate sampling pairwise tests to construct the descriptor. However, this construction involves a sequential memory access of as many steps as the binary string size. From the latter and motivated by the fact that modern computer vision tasks may require the construction of thousands, if not millions of binary descriptors, we propose to accelerate the construction process of the ORB descriptor via an FPGA-based hardware architecture. The latter is leveraged with a novel arrangement of pairwise tests, which takes advantage of a dual random access memory scheme achieving an acceleration of up to 17 times when compared against the sequential way. The empirical assessment indicates that ORB descriptors obtained from the proposed approach keep a similar performance to that of the original ORB.

The first author is supported by the Mexican National Council for Science and Technology (CONACyT) studentship number 627047. The second author is thankful for the support received through his Royal Society-Newton Advanced Fellowship with reference NA140454.

✉ Roberto de Lima
delima87@inaoep.mx

Jose Martinez-Carranza
carranza@inaoep.mx

Alicia Morales-Reyes
a.morales@inaoep.mx

Rene Cumplido
rcumplido@inaoep.mx

[1] Computer Science Department, Instituto Nacional de Astrofísica, Óptica y Electrónica, 72840 Puebla, Mexico

Keywords ORB · BRIEF · FPGA · Binary descriptors · Feature matching

1 Introduction

Based on a set of pairwise tests over pixel intensities, which form the descriptor's binary string, the binary robust independent elementary features (BRIEF) descriptor [7] and the first of its type proved to be an efficient and effective alternative to costly floating point-based descriptors already used in several computer vision tasks. Although it was experimentally shown that BRIEF was capable of offering reasonable discrimination capabilities, its way of describing an image patch was still in an early stage since there was not a clear way on how to choose the pairwise tests and there was a lack of robustness against rotation changes.

Since the introduction of BRIEF to the field, several approaches emerged aiming at bringing robustness to the pairwise tests against several image transformations. Among all the proposals, that presented by the oriented FAST and rotated brief (ORB) descriptor [30] has arguably gained a wide range of popularity mainly for two reasons: (1) It presents a learning-based scheme to choose the pairwise tests, aimed at selecting pair tests with a mean bit response close to 0.5 and with low correlation among the chosen pairs; (2) it utilizes a simple method based on image moments to estimate the dominant patch orientation, which can be used to rotate the pairwise tests, thus enabling robustness against rotation transformations.

In addition to the above, it is clear that the comparison operations, involved in the pairwise tests of ORB, are faster than those operations involved in the histogram construc-

Published online: 24 June 2017

tion of several floating point-based descriptors such as the scale-invariant feature transform (SIFT) or the histogram of oriented gradients (HoG) [10]. Yet, such comparison tests remain sequential as sequential memory access is required in the calculations. This sequential way of constructing ORB (or any other binary descriptor based on the pairwise tests) may seem negligible at first. However, with the advent of the high-definition format in video cameras, modern computer vision task has begun to demand the construction of thousands, if not millions, of descriptors. Hence, there is a genuine concern in reducing the processing time invested in the construction stage.

Motivated by the above, we propose a novel way of arranging the pairwise tests in order to accelerate the construction process of the descriptor. For the latter, we build upon our previous work [11], which is inspired by the flexibility of designing custom hardware, which can be achieved by using a field programmable gate array (FPGA). FPGAs have been used in the past to implement custom hardware architectures for SIFT [8,9,14,15,31], and speeded up robust features (SURF) [5,18,36,38], enabling these descriptors to run in real time. However, our approach is not a hardware architecture translation of the ORB descriptor, but a hardware implementation of the construction process, which is leveraged with our new way of arranging the pairwise tests such that data memory organization is exploited. The latter enables simultaneous readings of memory data, thus leading to a construction acceleration of up to 17 times when compared against the sequential way.

Although our novel approach proposes a new format of the pairwise tests, the core of the construction does not change since we are able to use ORB's learning scheme to chose the pairwise test, as much as the image moments-based method to detect the dominant orientation in order to rotate the pairwise tests accordingly. In order to assess our approach, we have carried out experiments via a software implementation and we have found that our improved ORB descriptor exhibits a similar performance to that of the original ORB against several image transformations that include rotation changes, but with the plus that our improved ORB descriptor is 17 times faster in terms of its construction process.

In order to describe our approach, the rest of the paper is organized as follows: In the next section, the related work is presented; Sect. 3 describes the ORB descriptor approach; Sect. 4 presents our proposed data memory organization and memory access scheme as well as its experimental evaluation; Sect. 5 presents a detailed description of our proposed FPGA architecture; the results obtained in our experiments are discussed in Sect. 6; and conclusions and future work are presented in Sect. 7.

2 Related work

Feature detection and description is a very active field of research in computer vision; hence, several local feature descriptors have been proposed in the literature in the last years. Many of those from SIFT onwards have used floating point or integer representations, such as SURF, PCA-SIFT [16], and KAZE [3], to name a few. These are widely used in a variety of applications as they are robust against visual transformations such as scaling and rotation. But recently binary descriptors such as BRIEF, ORB [30], BRISK [21], FREAK [1], NESTED [6], A-KAZE [2] emerged as an alternative to floating point descriptors, by offering low memory footprint, ease to compute, and a fast descriptor comparison, thus making binary features suitable to be the basic building block of many computer vision tasks, especially, given the rise of real-time and mobile-based applications.

BRIEF is one of the initial binary descriptors, proposed by Calonder et al. [7]. Basically, BRIEF takes a smoothed image patch around a keypoint (point of interest in a image) and it makes pixel intensities comparisons in order to construct a binary descriptor. Its performance is similar to floating point descriptors in many aspects, including robustness to lighting, blur, and perspective distortion. However, it is very sensitive to in-plane rotation.

From the above, Ethan Rublee et al. proposed a BRIEF descriptor that is invariant to orientation called ORB (oriented FAST and rotated BRIEF); the main contributions in this work lies in adding an orientation component to FAST [28] 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 [21] Leutenegger et. al proposed a binary descriptor invariant to rotation and scale. It uses the AGAST corner detector [22], which is an improvement in FAST. This binary descriptor is constructed by pixel comparison whose distribution forms a concentric circle surrounding the feature.

Based on human retina, in [1] a binary descriptor was proposed; it received the name of FREAK (Fast Retina Keypoint), and this is only a feature descriptor that heavily relies on a robust feature detector algorithm for keypoint detection; therefore, SURF and SIFT feature detectors are commonly used.

More recently, in [6] a new family of binary descriptors was proposed. By means of defining a set Hawaiian structure around a keypoint, and computing oriented gradients, the binary descriptor is computed. Moreover, a new local distance function, called the nesting distance, was introduced in order to remove most outliers.

In [2], Alcantarilla proposed an improvement in KAZE in terms of computational complexity and descriptors stor-

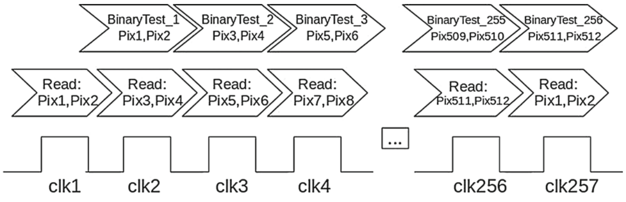


Fig. 1 Sequential process to construct a 256-bit binary descriptor

age, by using numerical schemes called fast explicit diffusion (FED) [12,34], and modifying the LDB descriptor [37]. This binary descriptor named A-KAZE is faster to compute than SIFT, SURF, and KAZE and exhibits a similar performance.

Notably, the use of binary features has been increased in recent years, thus leading several works reporting performance evaluations and comparisons among several types of binary descriptors. For instance, in [13,26], authors report studies for image matching and compare BRIEF, ORB, BRISK, SIFT, and SURF descriptors. More recently, aside from these latter descriptors, FREAK, NESTED, among others, are also compared in [17]. Experiment results point out that among binary descriptors, ORB is the best descriptor for feature matching, according to evaluation metrics proposed in [24]. In addition, they concluded that SIFT is still the most accurate performer for feature matching applications. However, KAZE descriptor is not compared, even though its performance is reported better than SIFT.

Since ORB features are well suited for feature matching application, and because their construction require a less complex process, different ORB hardware implementations have been proposed [4,19,20,33,35]. The latter proposed an FPGA implementation of the ORB algorithm, carrying out all integration steps by means of a pipeline strategy. However, those approaches require sequential memory access in order to construct the binary array in such a way that 257 clock cycles are needed to compute a 256-bit binary descriptor, see Fig. 1. The proposed FPGA-based descriptor construction process is sped up by exploiting parallelization from the novel arrangement of pairwise tests.

3 ORB descriptors

ORB features emerged from the need of making BRIEF descriptors invariant to orientation. Authors proposed to compute an orientation component, based on the dominant orientation in the patch, for each FAST keypoint. These FAST keypoints with orientation, also known as *oFAST* keypoints, are used to steer BRIEF w.r.t. the dominant orientation component [30]. Thus, for the sake of completeness, in this section a brief introduction to oFAST detector and BRIEF descriptor is provided.

3.1 oFAST

The FAST [28] salient point detector is widely used due to its performance and computational properties; however, FAST salient points do not have an orientation component. Therefore, in [30] a metric of corner orientation is used to make the FAST detector suitable to describe a feature invariant to orientation. This feature detector received the name of oFAST (oriented FAST).

The orientation component is calculated based on the *intensity centroid* [27]. Since corners intensity is offset from its center, the vector constructed between them may be used to induce orientation. According to Rosin [27], the centroid of a patch is defined as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{1}$$

where m represents the moments of a patch, defined as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \tag{2}$$

Then, a vector from the corner's center to the centroid is constructed in order to compute the patch orientation:

$$\theta = \operatorname{atan2}(m_{01}, m_{10}) \tag{3}$$

Next, BRIEF descriptor is detailed.

3.2 BRIEF

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:

$$\tau(p; x, y) := \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

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

$$f_{nd}(p) := \sum_{1<i\leq 1} 2^{i-1}\tau(p; x_i, y_i) \tag{5}$$

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. For the former, descriptors of 128, 256, and 512 bits proved to exhibit

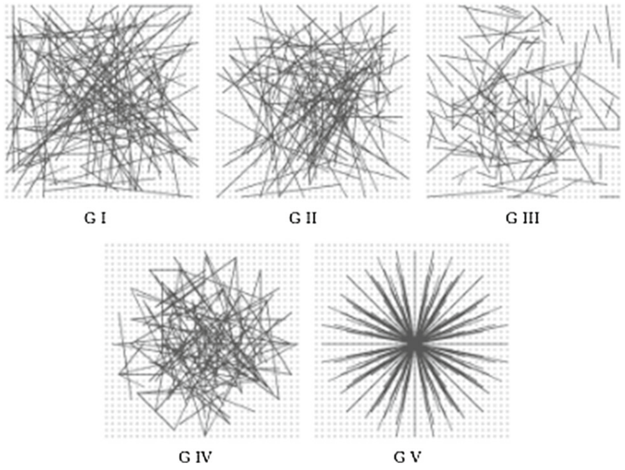


Fig. 2 Several explored binary tests distributions (images taken from [7])

good discrimination performance, with 256 being the current standard size [23,30]. On the other hand, many different types of distributions were considered in [7] for selecting *nd* test locations. Figure 2 taken from [7] shows the explored distributions, where experimental results reported that G III, a Gaussian BRIEF pattern, performs better.

In [30], authors explore different patterns, and they proposed a learning method with the aim of guaranteeing a set of discriminative binary tests. This method is described in the next subsection.

3.2.1 Learning good binary features

Rublee et al. [30] 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 [7], a typical Gaussian BRIEF pattern (see Fig. 2, G III) was reported as the best binary test distribution.

Considering this, a learning method for choosing a good set of binary tests was developed in [30]. Generically, the algorithm consists in extracting *m* keypoints from a set of images using FAST [28] or its variants [29]. 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.

4 Proposed pairwise tests arrangement

In this study, a memory access scheme to take advantage of image data allocation is proposed.

4.1 Image data allocation

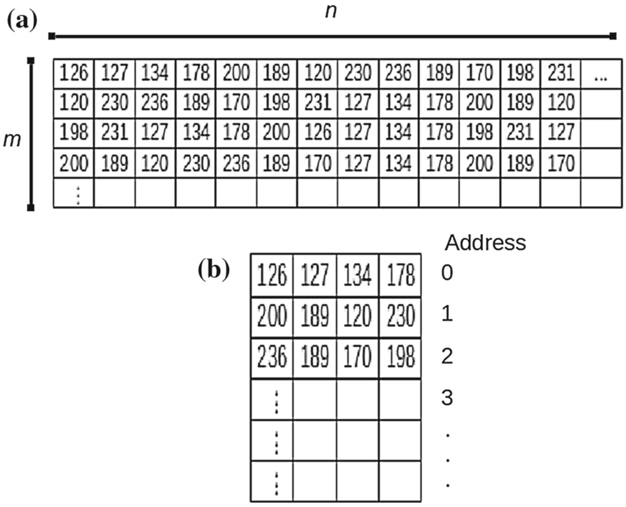
As we mentioned, ORB algorithm implies oFAST keypoints detection and steered BRIEF according to the orientation component. Therefore, in this research it is assumed that keypoints were previously computed and stored in a single-port RAM, and a gray 8-bit smoothed image is stored in a dual-port RAM.

RAM data can be allocated in 32-bit words; hence, a smoothed gray scale image can be stored in memory by clustering 4 pixels per memory address. Furthermore, in an FPGA hardware implementation, 8 image pixels can be simultaneously retrieved per clock cycle. Figure 3 clarifies the above mentioned.

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:

$$l = \left((x-1)\left(\frac{n}{4}\right)\right) + \frac{y-1}{4} \qquad (6)$$

On the other hand, corner keypoints found by oFAST method are allocated in a single-port RAM. The first 16 memory word bits refer 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 oFAST points detected, respectively.



126	127	134	178	200	189	120	230	236	189	170	198	231	...
120	230	236	189	170	198	231	127	134	178	200	189	120	
198	231	127	134	178	200	126	127	134	178	198	231	127	
200	189	120	230	236	189	170	127	134	178	200	189	170	
⋮													

126	127	134	178	0
200	189	120	230	1
236	189	170	198	2
⋮				3
⋮				.
⋮				.

Fig. 3 **a** Image with $m \times n$ resolution, **b** image memory organization

4.2 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 ORB descriptor, keeping both qualities explored in [30]: discriminative and uncorrelated binary tests.

Inspired by FREAK [1] and BRISK [21], which proved that pixels nearest to a keypoint offer relevant information, we propose to select 88 image pixels, around a salient point, grouped by 22 memory words located within a patch of 24 × 24 pixels, and also distributed in a foveal fashion, see Fig. 4. Note that 3 pixels are considered as an offset in order to deal with image patch boundaries.

From selected memory words, a set of binary tests is chosen by following the learning method described in Sect. 3, with a set of 2*k* FAST points extracted from images of the ZuBuD dataset [32]. Figure 5 shows a selection example of 32 out of 256 binary tests obtained according to the proposed learning method reported in [30].

In order to assess the binary test quality, the mean of every feature bit is calculated for all 2*k* samples. This measure is shown in Fig. 6, 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.

The proposed memory access scheme helps to construct a binary test distribution suitable to compute a high-quality ORB descriptor for discrimination while taking advantage of the retrieved image data. In the next subsection, the performance of this binary test distribution is evaluated for a feature matching application.

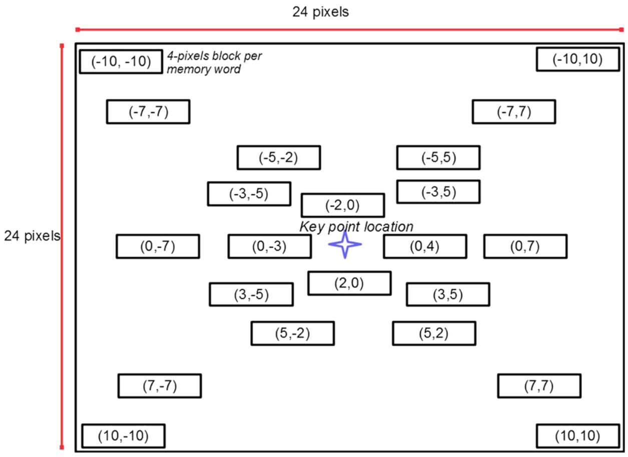


Fig. 4 Image memory words location for each keypoint

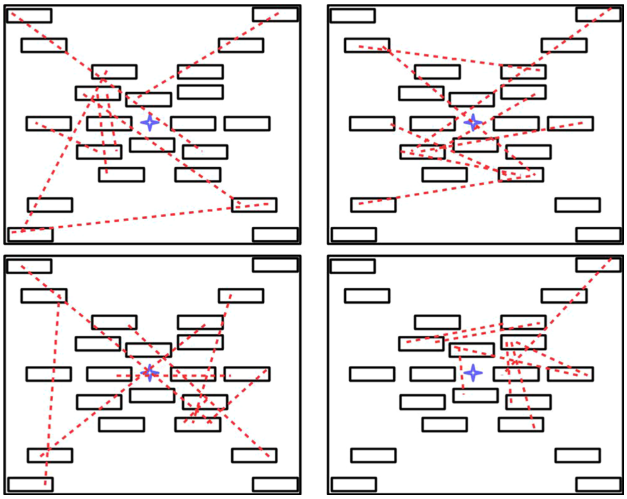


Fig. 5 Example of 32 binary tests via the learning method proposed in [30]

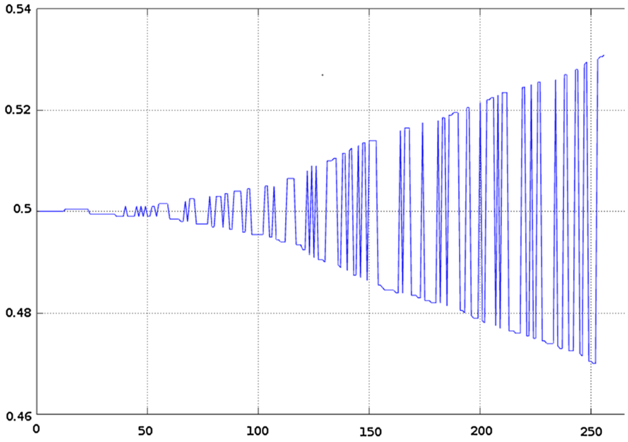


Fig. 6 Bit feature mean over 2k samples

4.3 Binary test distribution evaluation

The proposed binary test distribution is assessed using Mikolajczyk et al. evaluation method [24], using Oxford and Heinly datasets [13,25]. This method consists in evaluating a feature detector and a descriptor in a feature matching scenario. Thus, a oFAST and steered BRIEF blending with the proposed binary test distribution for feature detection and description, respectively, is defined.

Keeping in mind that the aim of feature matching lies in increasing the number of correct positives while minimizing the number of false positives, Mikolajczyk et al. [24] also proposed a metric called recall–precision; this criterion is based on the number of correct positives and the number of false positives obtained for an image pair, as follows:

$$recall = \frac{\text{Number of correct positives}}{\text{Total number of positives}} \tag{7}$$

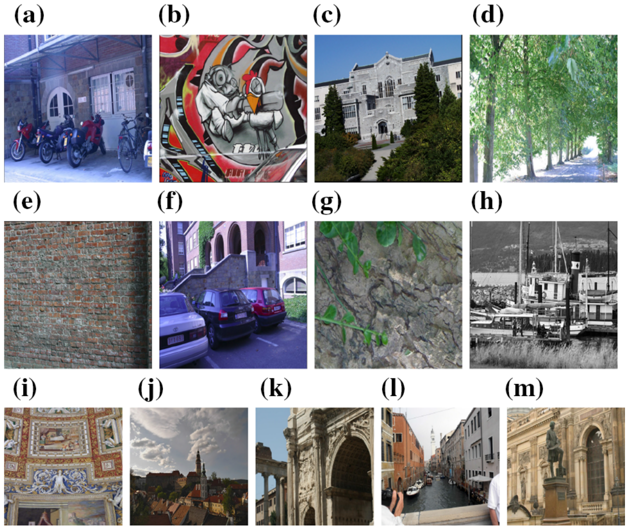


Fig. 7 Datasets used for evaluation. **a** Bikes, **b** Graffiti, **c** Ubc, **d** Trees, **e** Wall, **f** Leuven, **g** Bark, **h** Boat, **i** Ceiling, **j** DayNight, **k** Rome, **l** Venice and **m** Semper

and

$$1 - precision = \frac{\text{Number of false positives}}{\text{Number of matches (correct or false)}} \quad (8)$$

where the total number of positives is known a priori, determined from the ground truth homography provided by datasets. In this regard, 13 datasets are used for image matching estimation, and sample images from each dataset are shown in Fig. 7. Theses datasets cover different transformation such as view-point change (Graffiti and Wall), blur (Bikes and Trees), illumination changes (Leuven, DayNight), JPEG compression(Ubc), rotation (Bark Boat, Rome, Ceiling, Semper), and Zoom (Venice).

In order to assess the performance of ORB descriptors obtained via our novel construction scheme, a comparison against the following binary descriptors was carried out: ORB, BRISK, and A-KAZE (OpenCV 3.0 implementations). Figures 8, 9 and 10 depict quantitative results concerning the descriptor performance, considering a nearest neighbor matching strategy.

As illustrated in Fig. 8, ORB descriptor performance constructed via our approach is close to that of the original ORB descriptor, this is, for all affine transformations including rotation. This was expected as our approach uses the same binary test training method. Note that regarding illumination and rotation transformations datasets (Fig. 11), our approach is slightly behind from other approaches, see graphs (d) and (f), in Fig. 9. For illumination changes, as ORB descriptor construction relies on pixel intensities and pixel locations are restricted to carry out binary tests, the proposed approach performance is negatively affected. On the other hand, having a reduced number of features, such as in the bark dataset, also affects the proposed approach.

From the above, considering achieved performance results, the proposed approach is able to build high-quality ORB (or rotated BRIEF) descriptors, with performance comparable to that obtained with state-of-the-art binary descriptors.

In the next section, the proposed hardware architecture for acceleration of ORB descriptors calculation is described in detail.

5 Proposed architecture

In the previous section, we have proposed a memory access scheme suitable for accelerating the ORB descriptor construction via hardware, based on a defined pattern of binary tests which was chosen following the method proposed by Ethan R. et al. In addition, this binary test distribution was tested for an image matching application, obtaining comparable results to original ORB. Therefore, in this section we propose a parallel FPGA architecture for computing ORB descriptors with length of up to 256 bits. We focus on accelerating the binary tests calculation process. 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 Fig. 12, where oFAST 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 ORB descriptors, by steered BRIEF:

1. The first corner location is read by the address control.
2. The memory address of a 4-pixel block (see Fig. 4) is computed by the address generator block.
3. 4-pixel 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 Fig. 12 is provided.

5.1 Address control block

Address control block aims at synchronizing the entire process, as well as loading all 4-pixel blocks in the buffer. The core of this block lies in a Mealy state machine, see Fig. 13.

The Mealy state machine consists of five states. First, an address generator flag and a corresponding keypoint memory

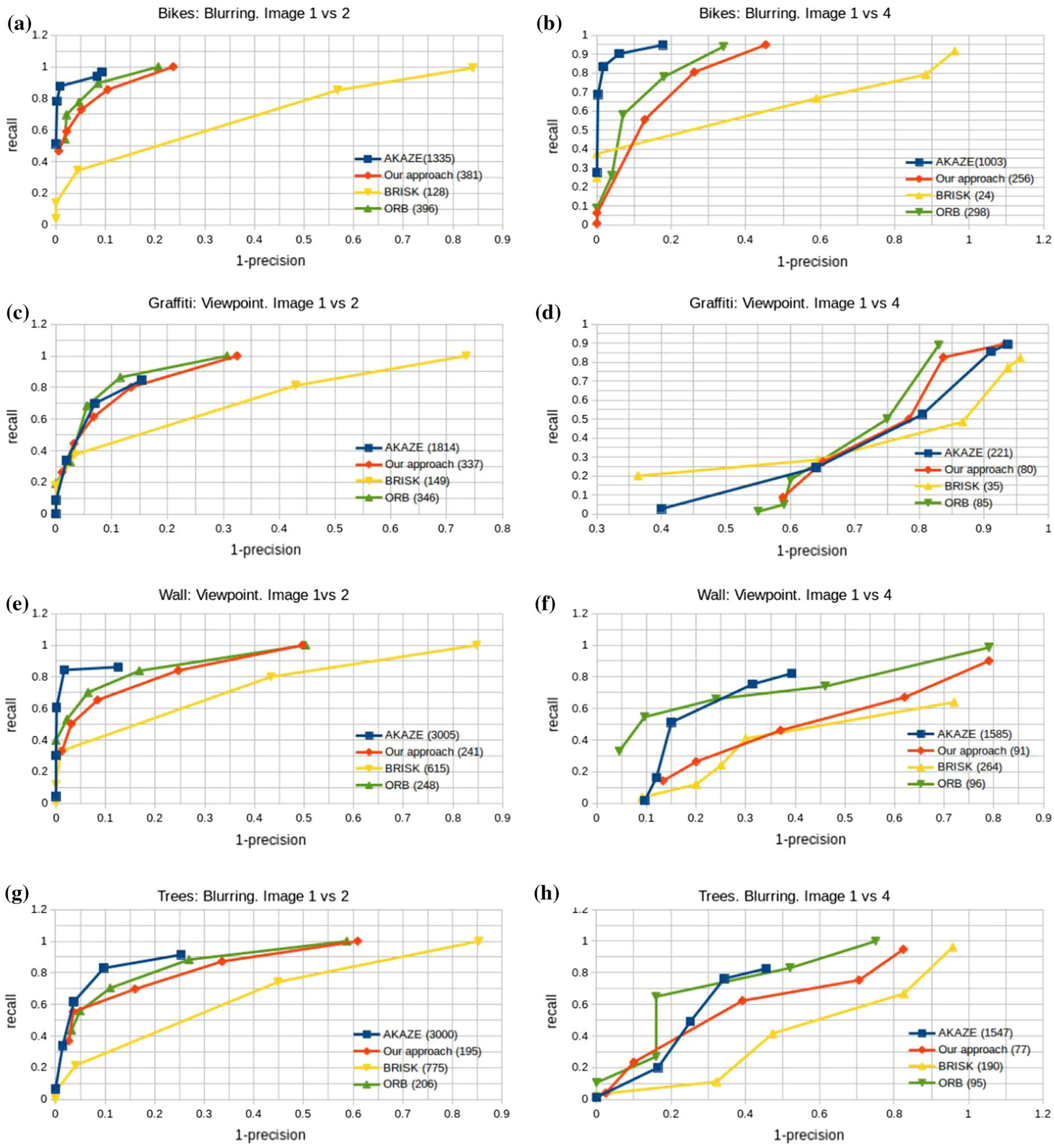


Fig. 8 Part 1. Transformations: view-point changes and blur. Recall versus 1 precision graphs for nearest neighbor strategy. *Curves* are obtained by tuning a threshold of distance between descriptors. In parentheses, next to the name of each of the methods, the number of found correspondences is shown. **a** Bikes, **b** Bikes, **c** Graffiti, **d** Graffiti, **e** Wall, **f** Wall, **g** Trees and **h** Trees

address are initialized in $S0$. After, a corresponding keypoint 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-pixel image blocks corresponding to current keypoint in the buffer. The dual-port RAM memory is synchronized with the address control block by a stall. Finally, in $S4$, two 4-pixel 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-pixel image blocks have been sent to the buffer. After that, the state machine

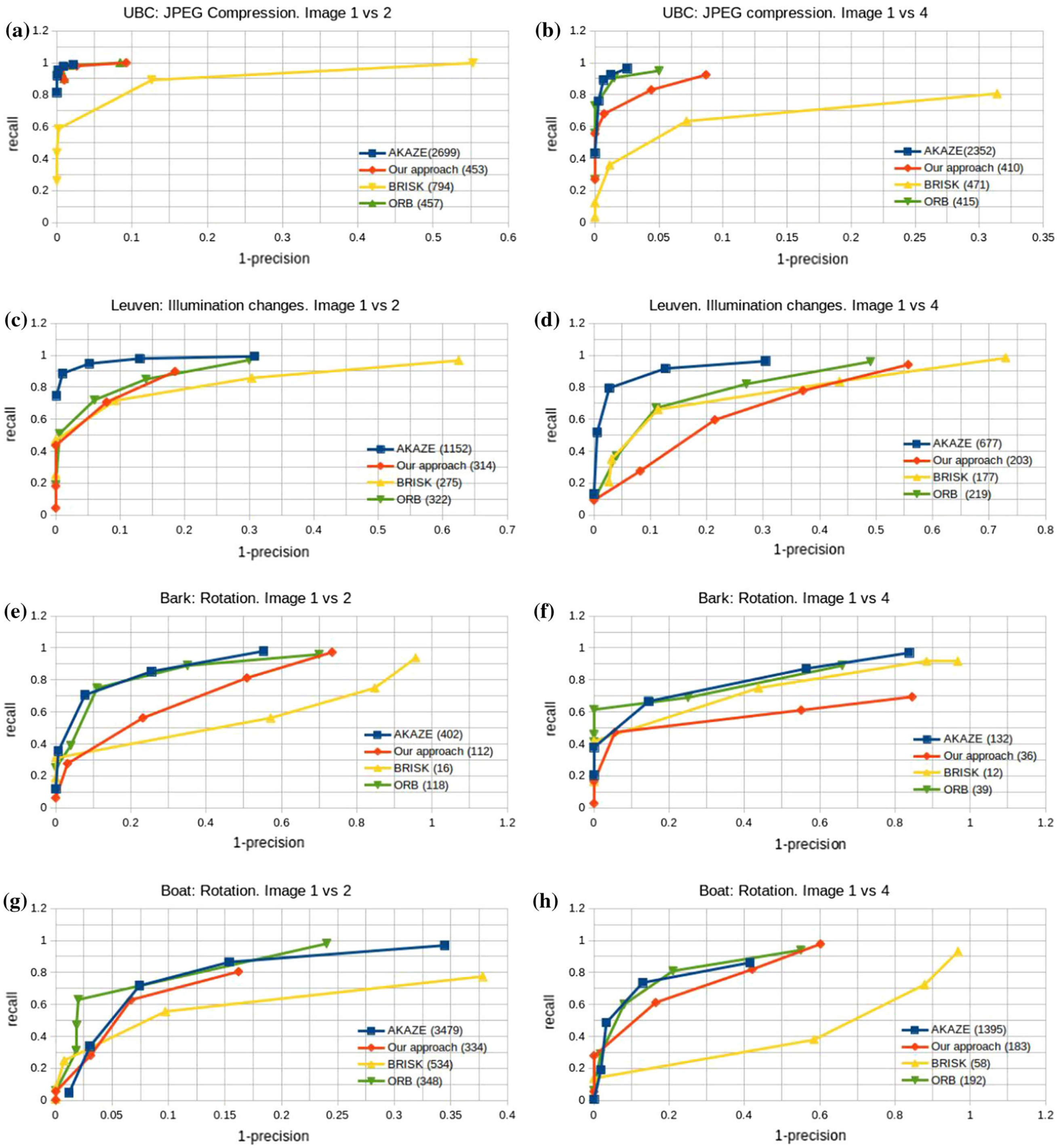


Fig. 9 Part 2. Transformations: JPEG compression, illumination changes and rotation. **a** UBC, **b** UBC, **c** Leuven, **d** Leuven, **e** Bark, **f** Bark, **g** Boat and **h** Boat

returns to $S0$ and the process is repeated for the next keypoints.

5.2 Address generator block

In Sect. 4, it has been demonstrated that only 22 memory locations corresponding to 4-pixel image blocks are enough for choosing a discriminative and uncorrelated set of 256 binary tests. Hereof, the memory address for each keypoint location is computed by the address generator block. Since a keypoint location is known a priori, the memory addresses showed in Fig. 3 are easy to compute by adding defined constants to (x,y) keypoint and following Eq. 3. It is important to mention that these constant values are determined by the

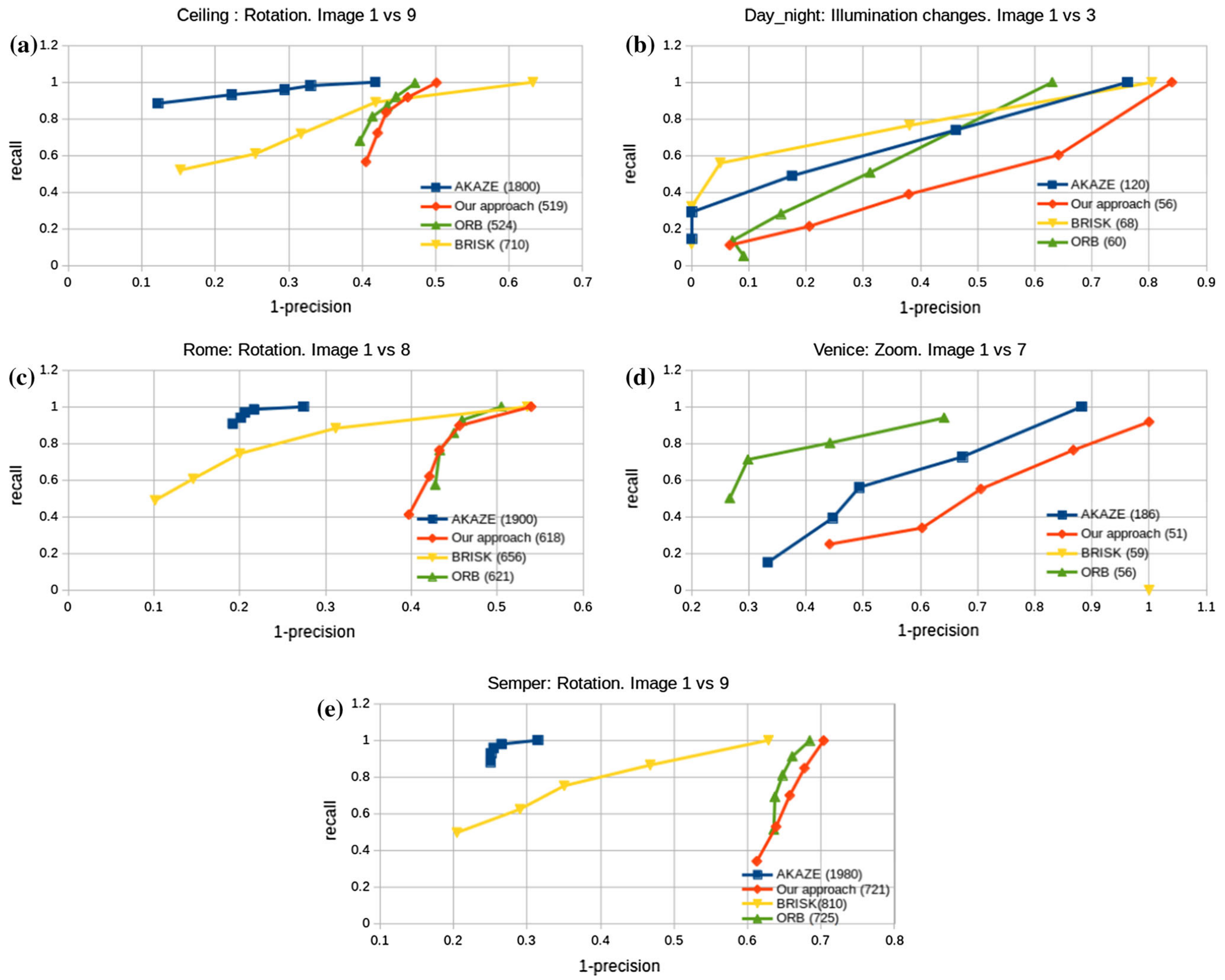


Fig. 10 Part 3. Heinly [13] dataset transformations: rotation, illumination changes and zoom. **a** Ceiling, **b** Daynight, **c** Rome, **d** Venice and **e** Semper

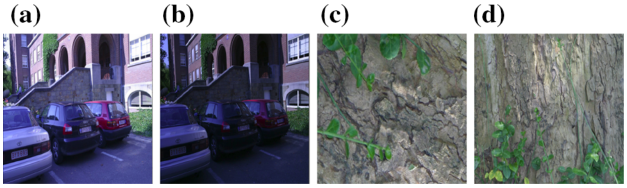


Fig. 11 Images with which the performance of our approach is surpassed. **a** Leuven 1, **b** Leuven 4, **c** Bark 1 and **d** Bark 4

address generator flag as well as they depend on orientation component.

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

5.3 Buffer and demux

A shift register is implemented with the purpose of temporarily storing all 4-pixel image blocks corresponding to each oFAST 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 Fig. 14). 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 Fig. 15. In consequence, once the buffer is filled, 21 demultiplexers are placed to split 22 4-pixel blocks in 88 pixels.

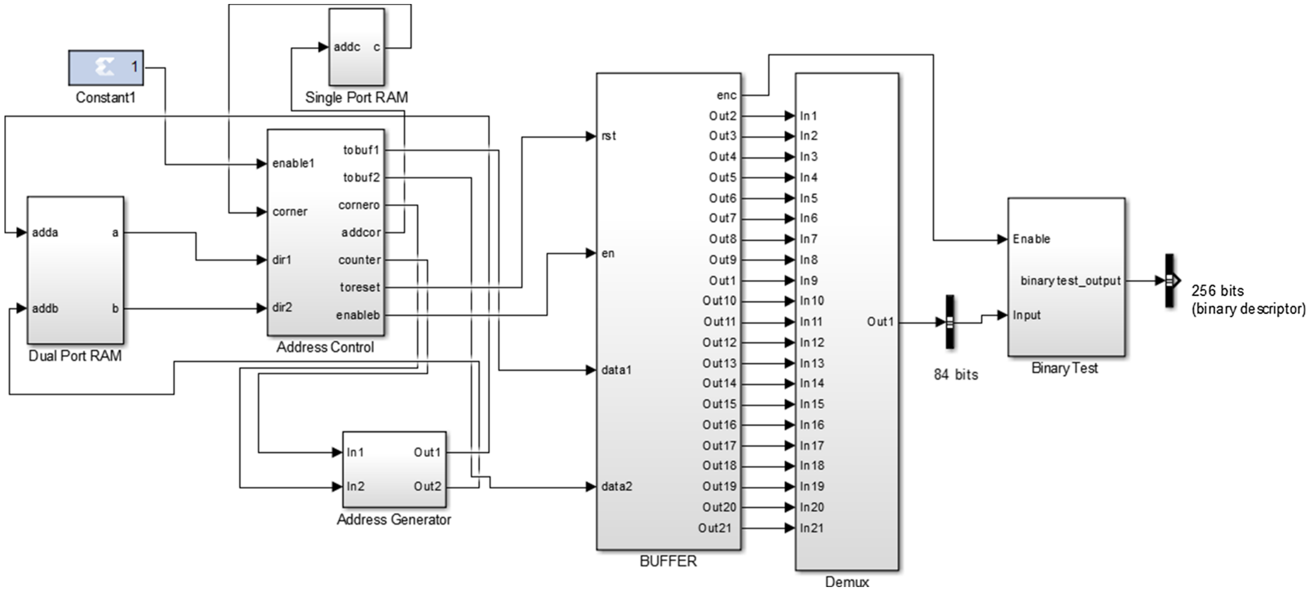


Fig. 12 FPGA architecture for the construction of ORB using our approach

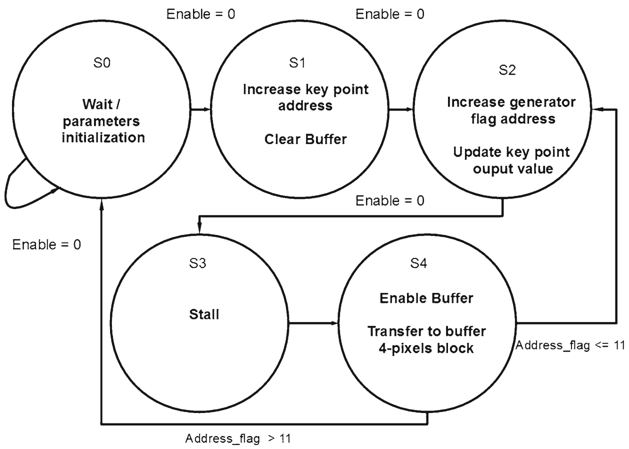

Enable = 0
Enable = 0
S0
Wait / parameters initialization
S1
Increase key point address
Clear Buffer
S2
Increase generator flag address
Update key point ouput value
Enable = 0
Enable = 0
S3
Stall
S4
Enable Buffer
Transfer to buffer 4-pixels block
Address_flag <= 11
Address_flag > 11

Fig. 13 Mealy state machine designed to model address control behavior

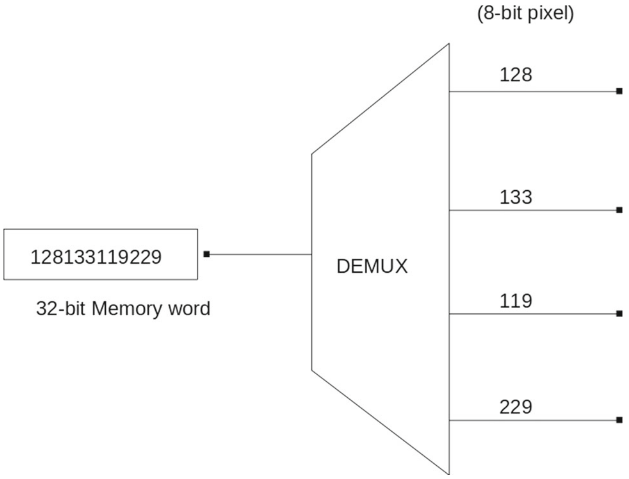


Fig. 15 32-bit to 8-bit word demultiplexer

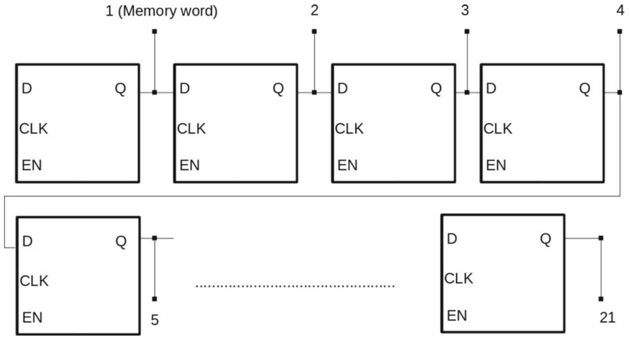


Fig. 14 Temporary storage via a shift register

5.4 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 Sect. 4. 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.

Figure 16 summarizes the whole process in the form of a pipeline diagram. Note that, in the first twelve clock cycles the 22 4-pixel blocks are loaded; then, they are split and wired to the binary test module where the construction of the 256-bit binary descriptor is carried out simultaneously.

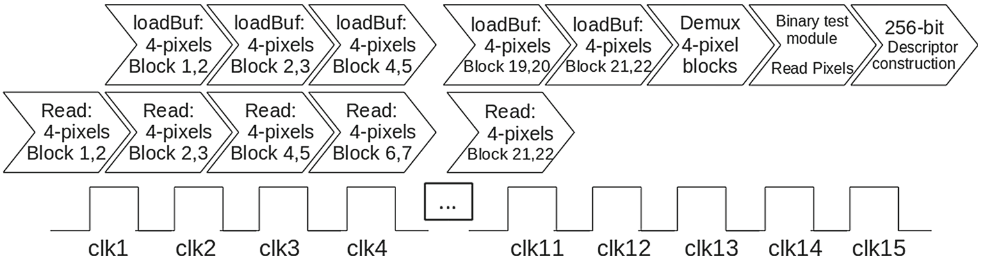

loadBuf: 4-pixels Block 1,2
loadBuf: 4-pixels Block 2,3
loadBuf: 4-pixels Block 4,5
loadBuf: 4-pixels Block 19,20
loadBuf: 4-pixels Block 21,22
Demux 4-pixel blocks
Binary test module Read Pixels
256-bit Descriptor construction
Read: 4-pixels Block 1,2
Read: 4-pixels Block 2,3
Read: 4-pixels Block 4,5
Read: 4-pixels Block 6,7
Read: 4-pixels Block 21,22
...
clk1
clk2
clk3
clk4
clk11
clk12
clk13
clk14
clk15

Fig. 16 Process to compute a 256-bit binary descriptor based on the proposed FPGA architecture

Table 1 Hardware resources

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

6 Results

The architecture is synthesized for Xilinx Zynq XC7Z020 SoC platform. A 320 × 240 smoothed image is stored in dual-port RAM, and 50 oFAST points are stored in a single-port RAM. Figure 12 gives an overview of the FPGA architecture which is assessed in terms of hardware resources and throughput. Table 1 shows the number of hardware resources used to compute a 256-bit ORB descriptor. A minimum programmable logic area is required for the whole architecture, resulting in a compact module. Calculating ORB 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 [20,35] , a sequential binary test is avoided. A 256-bit ORB descriptor is calculated in 15 clock cycles, in contrast to the 257 cycles taken by the sequential approach, thus accelerating the descriptor construction up to 17 times. Therefore, since clock frequency is 125 MHz, for the synthesized device, the proposed hardware architecture is capable of computing ORB descriptors of 50 keypoints in 6 ms.

The most important highlights from previous results are 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, and (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 ORB descriptor, while preserving descriptor quality. From this analysis, our work is highly motivated by the scenario where an image is preprocessed and oFAST points previously computed by a computer processing unit using OpenCV or by a module designed in Vivado HLS using Xilinx video libraries. 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 ORB descriptors.

7 Conclusion

We have presented a novel FPGA-based arrangement of pairwise tests used in the construction of 256 bits ORB descriptors. This arrangement is advantageous when using a dual random access memory in the FPGA, which enables a reduction in the number of memory accesses involved in their construction, thus enabling a parallel processing that accelerates the construction process up to 17 times when compared to those FPGA architectures whose acceleration relies on a pipeline strategy. In addition, considering a sequential implementation of the ORB descriptor construction on a CPU, it took 768 clock cycles to compute a 256-bit binary vector, in that case the achieved acceleration is around 51 times. Furthermore, this new arrangement follows the learning-based methodology presented by Rublee et. al. [30] in order to select good pairwise tests, under our proposed arrangement, such that these tests maintain a mean bit response close to 0.5 and low correlation among them, suggested conditions for the binary string to exhibit good discrimination capabilities.

Regarding the performance of the ORB descriptor constructed with our approach, in terms of discrimination capabilities, we have also presented a thorough software-based evaluation, following the assessment criterion presented by Mikolajczyk and Schmid [24]. The results indicate that the ORB descriptor constructed with our approach performs very closely to the original ORB under several image transformations, i.e., view-point changes, blurring, illumination changes, JPEG compression, zoom, and rotation. We should highlight that even in those cases where our constructed ORB descriptor falls slightly behind the original ORB, in terms of precision and recall, the difference may be deemed acceptable for the sake of accelerating the construction process up to 17 times; this of course will depend on the vision task or application.

In addition to the experimental assessment of our approach, we have also presented a detailed description of the FPGA implementation, including a report on hardware resources utilized in the FPGA. Note that architecture described in this work can be scaled seamlessly to compute a binary string of 512 bits. This can be achieved by sampling more pairwise test from the new arrangement presented in this work, with the caveat that the more pairwise tests are sampled, over the current new arrangement, the less low correlation among these will be guaranteed, as we have experimentally observed. How to overcome this is, however, an issue that will be dealt in our future work. In this sense, we have already begun to look at some options, for instance, instead of using predefined locations (based on a foveal distribution) of the 4-pixel image blocks, we will implement a search over all the possible groups of 4 pixels in the patch, including overlapping.

References

1. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
2. Alcantarilla, P.F.: TrueVision solutions: fast explicit diffusion for accelerated features in nonlinear scale spaces. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1281–1298 (2013)
3. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: Kaze features. In: Computer Vision–ECCV (2012)
4. Bello, E.D., Salvadeo, P.A.: An image descriptors extraction hardware–architecture inspired on human retina. In: 2014 IX Southern Conference on Programmable Logic (SPL)
5. Bouris, D., Nikitakis, A., Papaefstathiou, I.: 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 IEEE, pp. 3–10 (2010)
6. Byrne, J., Shi, J.: Nested shape descriptors. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1201–1208 (2013)
7. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: binary robust independent elementary features. Comput. Vis. ECCV **2010**, 778–792 (2010)
8. Chang, L., Hernandez-Palancar, J.: A hardware architecture for sift candidate keypoints detection. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 95–102. Springer (2009)
9. Chang, L., Hernandez-Palancar, J., Sucar, L.E., Arias-Estrada, M.: Fpga-based detection of sift interest keypoints. Mach. Vis. Appl. **24**(2), 371–392 (2013)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on IEEE, vol. 1, pp. 886–893 (2005)
11. de Lima, R., Martinez-Carranza, J., Morales-Reyes, A., Cumplido, R.: Accelerating the construction of brief descriptors using an fpga-based architecture. In: 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig), pp. 1–6. IEEE (2015)
12. Grewenig, S., Weickert, J., Bruhn, A.: From box filtering to fast explicit diffusion. In: Pattern Recognition, pp. 533–542. Springer (2010)
13. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Computer Vision–ECCV 2012, pp. 759–773. Springer (2012)
14. Huang, F.-C., Huang, S.-Y., Ker, J.-W., Chen, Y.-C.: High-performance sift hardware accelerator for real-time image feature extraction. Circuits Syst. Video Technol. IEEE Trans. **22**(3), 340–351 (2012)
15. Jiang, J., Li, X., Zhang, G.: Sift hardware implementation for real-time image feature extraction. Circuits Syst. Video Technol. IEEE Trans. **24**(7), 1209–1220 (2014)
16. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on IEEE, vol. 2, pp. II–II (2004)
17. Khan, N., McCane, B., Mills, S.: Better than sift? Mach. Vis. Appl. **26**(6), 819–836 (2015)
18. Krajník, T., Šváb, J., Pedre, S., Čížek, P., Přeučil, L.: Fpga-based module for surf extraction. Mach. Vis. Appl. **25**(3), 787–800 (2014)
19. Lee, K.-Y.: A design of an optimized orb accelerator for real-time feature detection. Int. J. Control Autom. **7**(3), 213–218 (2014)
20. Lee, K.-Y., Byun, K.-J.: A hardware design of optimized orb algorithm with reduced hardware cost. Adv. Sci. Technol. Lett. **43**, 58–62 (2013)
21. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: binary robust invariant scalable keypoints. In: Computer Vision (ICCV), 2011 IEEE International Conference on IEEE, pp. 2548–2555 (2011)
22. Mair, E., Hager, G., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Computer Vision–ECCV 2010, pp. 183–196. Springer (2010)
23. Martinez-Carranza, J., Mayol-Cuevas, W.: Real-time continuous 6d relocalisation for depth cameras. In: Workshop on Multi VIew Geometry in RObotics (MVIGRO), in Conjunction with RSS (2013)
24. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Pattern Anal. Mach. Intell. IEEE Trans. **27**(10), 1615–1630 (2005)
25. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**(1–2), 43–72 (2005)
26. Mukherjee, D., Wu, Q.J., Wang, G.: A comparative experimental study of image feature detectors and descriptors. Mach. Vis. Appl. **26**(4), 443–466 (2015)
27. Rosin, P.L.: Measuring corner properties. Comput. Vis. Image Underst. **73**(2), 291–307 (1999)
28. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Computer Vision–ECCV 2006, pp. 430–443. Springer (2006)
29. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. Pattern Anal. Mach. Intell. IEEE Trans. **32**(1), 105–119 (2010)
30. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on IEEE, pp. 2564–2571 (2011)
31. Shao, A.J., Qian, W.X., Gu, G.H., Lu, K.L.: 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 (2015)
32. Shao, H., Svoboda, T., Van Gool, L.: Zubud-zurich buildings database for image based recognition. In: Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland. Technical Report, vol. 260, p. 20 (2003)

33. Viswanath, P., Swami, P., Desappan, K., Jain, A., Pathayapurakkal, A.: Orb in 5 ms: an efficient SIMD friendly implementation. In: Computer Vision-ACCV 2014 Workshops, pp. 675–686. Springer (2014)
34. Weickert, J., Grewenig, S., Schroers, C., Bruhn, A.: Cyclic schemes for PDE-based image analysis. Int. J. Comput. Vis. 1–25 (2015)
35. Weberruss, J., Kleeman, L., Drummond, T.: Orb feature extraction and matching in hardware. In: Australasian Conference on Robotics and Automation (2015)
36. Wilson, C., Zicari, P., Craciun, S., Gauvin, P., Carlisle, E., George, A., Lam, H.: A power-efficient real-time architecture for surf feature extraction. In: ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on IEEE, pp. 1–8 (2014)
37. Yang, X., Cheng, K.T.: LDB: an ultra-fast feature for scalable augmented reality on mobile devices. In: Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on IEEE, pp. 49–57 (2012)
38. Zhao, J., Zhu, S., Huang, X.: Real-time traffic sign detection using surf features on fpga. In: High Performance Extreme Computing Conference (HPEC), 2013 IEEE, pp. 1–6 (2013)

Roberto de Lima is a Research Assistant in the Computer Science Department at Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE). He obtained a B.Sc. degree in Electronic Engineering from the Benemérita Universidad Autónoma de Puebla and a M.Sc. degree in Computer Science from INAOE, both institutions in Mexico. His research interests lie in computer vision and high-performance computing.

Jose Martinez-Carranza is an Associate Professor in the Computer Science Department and member of the Robotics Laboratory at the Instituto Nacional de Astrofísica, Óptica y Eletrónica (INAOE). He obtained a B.Sc. in Computer Science (Cum Laude) from the Benemérita Universidad Autónoma de Puebla in 2004 and an M.Sc. in Computer Science (Best Student) from INAOE in 2007, both institutions in México. In 2012, he received his Ph.D. from the University of Bristol in the UK, where he also worked as Research Assistant and Associate from 2012 to 2014. His interest includes robotics, computer vision, machine learning applications, and high-performance computing. He has received the highly prestigious *Newton Advanced Fellowship* (2015–2018), granted by the Royal Society in the UK. In 2016, he led a team and won the *second place* in the *International Micro Air Vehicle Competition (IMAV 2016)*, indoors category; and in 2017, the *first place* in the *Autonomous Drones Category—Advanced Level* of the Mexican Robotics Tournament.

Alicia Morales-Reyes was admitted to the Ph.D. degree in the College of Science and Engineering at the University of Edinburgh in 2011. She developed her research within the system level integration group at the Institute for Integrated Micro and Nano Systems. In 2006, she received the M.Sc. degree in Computer Science from the National Institute for Astrophysics, Optics and Electronics in Tonantzintla, Mexico. She obtained a BEng in Electrical and Electronics Engineering from the National Autonomous University of Mexico, in 2002. Currently, she is a titular researcher in the Computer Science Department at the National Institute for Astrophysics, Optics and Electronics. She collaborates within the Reconfigurable and High Performance Computing research group.

Rene Cumplido is a professor at the Computer Science Department at INAOE. He holds a B.Sc. degree in Computer Science and a M.Sc. degree in Electrical Engineering from CINVESTAV. In 2001, he received the Ph.D. degree in Electrical Engineering from Loughborough University, UK. He is co-founder and chair of International Conference on Reconfigurable Computing and FPGAs, ReConFig. He is the founder and served as editor-in-chief (2007–2011) of the International Journal of Reconfigurable Computing, IJRC. He is active in a number of technical committees of international conferences and has served as associate and guest editor in several international journals. His research interests are reconfigurable computing and custom architectures.