This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 1 |

**Ground Control Point Automatic Extraction for Spaceborne Georeferencing Based on FPGA**

**Dequan Liu, Guoqing Zhou\*, Dianjun Zhang, Xiang Zhou, and Chenyang Li**

***Abstract*—Feature points that are obtained from the combinedspeeded-up robust feature (SURF) detector and binary robust independent elementary features (BRIEF) descriptor have a highly robust performance. These points are previously considered the ground control points (GCPs) for building a connection between the image coordinates and the corresponding geodetic coordinates. This paper proposes a novel architecture to automatically and intelligently extract GCPs based on field programmable gate arrays (FPGAs). The parallelization SURF detector, BRIEF descriptor and BRIEF matching are implemented in a single Xilinx XC7VX980T FPGA system. Word length reduction (WLR), memory-efficient parallel architecture (MEPA), shift and subtraction strategies (SAS), a sliding window for separable convolution, and an optimized multispacer-scale are used to optimize the SURF detector. Improved parallel adder trees are used to accelerate the BRIEF matching. The proposed system achieves 380 frame per second (fps) with a 100 MHz clock frequency, which satisfies the real-time and low-power requirements of embedded devices. The results of the experiment demonstrate that the proposed architecture, when mapped onto a Xilinx Virtex-7 XC7VX980T FPGA device, can select the robust feature points.**

***Index Terms*—Binary Robust Independent ElementaryFeatures (BRIEF), Field Programmable Gate Arrays (FPGAs), Ground Control Points (GCPs), Georeferencing, Speeded-Up Robust Feature (SURF).**

1. INTRODUCTION

Georeferencing is an important technique of remote sensing (RS) image for an extensive variety of tasks, and estimation of the mathematical geometric calibration function of RS image based on the ground control points (GCPs) is

This work was supported in part by the Natural Science Foundation of China under Grant numbers: 41431179, 41601365,4143379; GuangXi Key Laboratory for Spatial Information and Geomatics Program (Contract No.17-259-16-09); GuangXi innovative Development Grand Grant under the grant numbers: GuiKe AA18118038, GuiKeAA18242048; the National Key Research and Development Program of China under Grant numbers 2016YFB0502501 and the BaGuiScholars program of Guangxi (Guoqing Zhou).

D. Liu and X. Zhou are with the School of Microelectronics, Tianjin University, Tianjin 300072, China (e-mail: 1017232001@tju.edu.cn, zqx0711@tju.edu.cn).

\*G. Zhou is with the School of Microelectronics, Tianjin University, Tianjin 300072, China, and GuangXi Key Laboratory for Spatial Information and Geomatics, Guilin University of Technology, Guilin, Guangxi, 541004, China. (e-mail: gzhou@glut.edu.cn).

D. Zhang and C. Li are with the School of Marine Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: zhangdj@tju.edu.cn, lichenyang\_1008@tju.edu.cn).

necessary. Traditionally, GCPs are obtained by special equipment or manually selected from a reference image or topographic, which is computationally expensive. Furthermore, GCPs cannot satisfy the practical real-time performance [1-3] of RS image processing. The primary task of the RS image matching is to find the correct GCPs correspondences on the reference image and the warped image [4]. Therefore, an automatic method for selecting GCPs is desired. This paper aims to automatically extract GCPs between the remotely sensed image and the corresponding reference image in real-time.

In image matching, the keypoint can be recognized between two or more images of the same scene. The images can be employed in frequent matching for different times, perspectives, and scales [5]. The local invariant feature has an important role in image matching. The literature on local feature detection is vast and dates to 1954, when Attneave [6] observed that information about shape was concentrated at dominant points with high curvature [7]. In 1980, Moravec [8] proposed a detector that was repeatable for small variations and near edges and was applied for stereo image matching. Harris and Stephens [9] improved the prior Moravec detector in 1988. The Harris corner detector includes gradient information and eigenvalues of symmetric positive, which are defined as a 2×2 matrix to make it more repeatable. The Harris corner detector is a prevalent feature detection technique that combines a corner detector and edge detector based on a local autocorrelation function. However, it is not scale-invariant and sensitive to noise [10]. Smith and Brady [11] developed the smallest univalue segment assimilating nucleus (SUSAN) detector in 1997. SUSAN is not sensitive to local noise and high anti-interference ability [4]. To obtain the scale-invariant feature, Lindenberg [12-14] investigated the scale-invariant theory and presented a framework for selecting local appropriate scales that can be used for automatic scales selection. Mikolajczyk *et al.* [15] presented the Harris-Laplace and Harris-Affine detectors and gradient location and orientation histogram (GLOH) detector [16]. David Lowe [17] proposed the scale-invariant feature transform (SIFT) algorithm. Since each layer relied on the previous layer and images had to be resized, it was not computationally efficient

1. Many SIFT variants, such as PCA-SIFT [19], GSIFT [20], CSIFT [21], and ASIFT [22], are relatively highly efficient. The Speeded-Up Robust Features (SURF) is a local descriptor that is inspired by SIFT. SURF was first introduced in [23] and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 2 |

fully explained in [24].

Recently, binary descriptors are being developed for image processing. The binary robust invariant scalable keypoints (BRISK) detector is a novel method for keypoint detection, and description, and matching method that was proposed by Leutenegger *et al.* [25]. BRISK is constructed by pixel comparisons, whose distribution forms a concentric circle surrounding the feature. Calonder *et al.* [26] investigated the binary robust independent elementary features (BRIEF) to efficiently extract features. The BRIEF descriptor vector consists 512-, 256- and 128-bit vectors. Hence, this feature substantially reduces the memory required to store the feature descriptor and the time consumed to match the features, while yielding comparable recognition accuracy [27]. Rublee *et al.*

1. proposed a very fast binary descriptor, named ORB, which was rotation invariant and resistant to noise. The main contribution in this work is the addition of an orientation component to the feature from accelerated segment test (FAST)
2. feature detector and to propose a learning method for choosing pairwise tests with excellent discrimination power and a low correlation response among the tests [30]. In [31], Alahi *et al.* suggested the fast retina keypoint (FREAK) as a fast compact and robust keypoint descriptor.

Feature-based matching algorithms have been extensively employed for a variety of applications, such as object localization, object recognition, motion estimation, and 3D reconstruction. Although, these algorithms have higher image matching performance and precision. They cannot achieve a performance that is sufficiently high to satisfy practical real-time requirements due to the computational complexity and vast memory consumption [32-35]. Therefore, researchers have applied the matching algorithm to real-time applications. These studies can be grouped into two main approaches: The first approach aims to reduce the complexity of the matching algorithm without losing precision. A principal component analysis (PCA) [18] and linear discriminant analysis (LDA) [36] are dimensionality reduction techniques that reduce the size of the original descriptor, such as SIFT or SURF [26, 37]. Calonder *et al.* [38] proposed a concept that used a shorten descriptor to quantize its floating-point coordinates into integers codes on fewer bits; the same result was proposed in [39, 40]. This concept is an effective way to replace the original complex detector or descriptor with a speeded-up detector or binary description, such as FAST [29], BRISK [25], BRIEF [26], ORB [28], and FREAK [31]. Lowe [15] approximated the Laplacian of Gaussian (LoG) via the difference of Gaussians (DoG) filter. Bay *et al.* [23] proposed approximation to the LoG by using box filter representations of the respective kernels based on SIFT.

The second method is focused on improving the processing speed using dedicated hardware such as multicore central processing units (CPUs), graphic processing units (GPUs), application specific integrated circuits (ASICs) and field programmable gate arrays (FPGAs). Čížek *et al.* [41] proposed a processor-centric FPGA-based architecture for a latency reduction in the vision-based robotic navigation. Krajník *et al.*

1. presented a complete hardware and software solution of an

FPGA-based computer vision embedded module that can carry out the SURF image feature extraction algorithm. Yao *et al.* [43] proposed an architecture of optimized SIFT feature detection for an FPGA implementation of image matching. The total dimension of the feature descriptor had been reduced to 72 from 128 of the original SIFT. In [44], a parallelization and optimization method to effectively accelerate the SURF was proposed and achieved a maximum of 83.80fps in a real-machine experiment, which enables real-time processing. Cheon *et al.* [45] analyzed the SURF algorithm and presented a fast descriptor extraction method that eliminated redundant operations in the Haar wavelet response step without additional resources. Kim *et al.* [46] presented a parallel processing technique for real-time feature extraction in object recognition by autonomous mobile robots, which utilized both CPU and GPU by combining OpenMP, Streaming SIMD Extension (SSE) and CUDA programming. Schaeferling *et al.* [47] described two embedded systems (ARM-based microcontroller and intelligent FPGA) for object detection and pose estimation using sophisticated point features. The feature detection step of the SURF algorithm was accelerated by a special IP core. Huang *et al.* [48] designed an architecture that combined the FAST detector and the BRIEF descriptor for detection and matching with subpixel precision. Lima *et al.* [30] proposed a hardware architecture based on the BRIEF descriptor. This approach contributed to reducing the number of memory accesses required to obtain the descriptor while maintaining its discrimination quality. Zhao *et al.* [49] presented an efficient real-time FPGA implementation for object detection. The system employed the SURF algorithm to detect keypoints in every video frame and applied the FREAK method to describe the keypoints. In [50, 72], a modified SURF detector and BRIEF descriptor matching algorithm based on FPGA were presented. To accelerate the SURF algorithm, an improved FAST feature point combined with the SURF descriptor matching algorithm was proposed in [51], which realized the real-time matching of target images.

Inspired by previous research, this paper proposes a hardware architecture to automatically extract GCPs for the RS image that is georeferenced based on the FPGA, which significantly reduces the computational requirement. The main contribution of this paper is considered as follows:

1. A parallelization SURF detector and BRIEF descriptor are designed in an FPGA.
2. Five approaches are applied to optimize the SURF detector for hardware implementation. Moreover, a novel suppressed method for candidate points that are not local maxima in the three-dimensional scale-space domain is suppressed zero in the non-maximal suppression step, which ensures that feature points do not overlap and are evenly spread over the input image.
3. The BRIEF descriptor is adopted in the proposed system, whose memory footprint is only a 256-bit vector, i.e., 32-byte. However, the original SURF (O-SURF) descriptor is a 512-bit vector of floating points, representing it still requires a 64-byte. Moreover, the implementation of the BRIEF matching is highly efficient in an FPGA, which has the minimum Hamming

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 3 |

distance between the reference image and the sensed image. Improvement adder trees are employed to reduce the complexity of the BRIEF matching step.

The remainder of the paper is organized as follows: Section

1. provides an overview of the SURF detector and BRIEF descriptor. Five approaches for optimizing SURF feature detection is introduced in Section III. Section IV describes the parallel implementation on the SURF detector, the BRIEF descriptor and the BRIEF matching, which are implemented in a Xilinx XC7VX980T FPGA. The evaluation experimental results and comparison with existing schemes are reported in Section V. Section VI concludes this paper.
2. FEATURE DETECTOR AND DESCRIPTOR ALGORITHM

The process of feature-based matching has three steps: feature detection, feature description, and feature matching. In this paper, an efficient real-time FPGA-based system which comprises SURF feature detection, BRIEF descriptor, and BRIEF matching, in a single chip, is investigated. The diagram of the proposed process is presented in Fig. 1. The feature detection and feature description module automatically extract GCPs from the RS image. The feature matching module automatically matches the image pairs based on the ground control point [4].

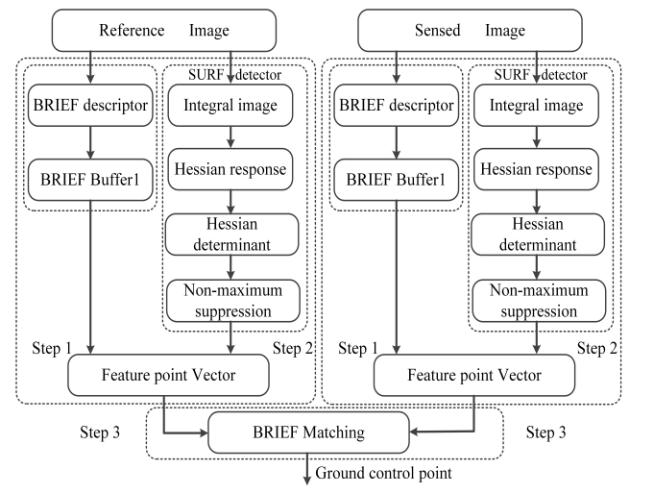


Fig. 1. Diagram of ground control point extraction for the remote sensing images.

*A. SURF Feature Detector*

SURF [23, 24] is scale- and rotation-invariant, which takes a grayscale image as an input. As shown in Fig. 1, SURF detector can be divided into three steps, Integral image, Hessian response, and non-maximum suppression. SURF feature detector will be briefly summarized in this section.

*1) Integral Image*

Integral image is a novel method to improve the performance of the subsequent steps of SURF detector [30]. The integral image is used as a rapid and effective way to calculate summations over image subregions [24]. Given the pixel value *i* ( *x* , *y*) for the coordinates ( *x* , *y*) of an image with width *W*

and height *H*, the value of coordinates ( *x* , *y*) in the integral

image *ii* ( *x* , *y*) can be defined as [23, 52]:

|  |  |
| --- | --- |
| *xy* |  |
| *ii* (*x*, *y* )=*i* (*x* ', *y* ') 0 *x W* ,0 *y H* | (1) |

*x* '=0 *y* '=0

Through the concept of integral image (as shown in Fig. 2), the cumulative sum of all the pixels in the rectangular region with the coordinates ( *x* , *y*) of the top-left pixel, width *w*, and

height *h* can be stated mathematically as in [53]

|  |  |  |
| --- | --- | --- |
| *x* + *w*−1 *y* + *h* −1 |  |  |
| *Sw* ,*h* (*x*, *y*)=*i*(*x* ', *y* ')= *ii*(*x* −1, *y* −1)+ |  |  |
| *x* '= *x y* '= *y* | (2) |  |
| *ii*(*x* + *w* −1, *y* + *h* −1)− *ii*(*x* −1, *y* + *h* −1) |  |
|  |  |
| −*ii* (*x* + *w* − 1, *y* + *h* −1) |  |  |
| the initial condition of (2) is |  |  |
| *ii* (−1, *y* )= *ii* ( *x* ,−1)= *ii*(−1,−1)=0 | (3) |  |

According to (2), integral image provides a fast way to get the sum histogram of an arbitrary-sized rectangle, requiring only three adders and calculating near-constant-time.

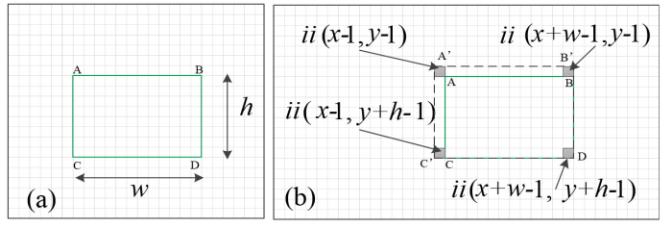


Fig. 2. Calculation cumulative sum of all pixels in the rectangular region by (2). With an integral image, the sum values of the green rectangular region that is bounded with A, B, C, and D are calculated by *ii* ( *A* ') + *ii* ( *D*) − *ii* (*C* ') − *ii* ( *B* ')

*2)* *Hessian Response*

In the O-SURF [23, 24], the scale space was established by the purpose of the scale invariance. The scale space can be divided into *o* ( *o* 1) octaves, and each octave is further divided

into *v* ( *v* 3) intervals to obtain a total of *o* *v* scale-levels.

Each interval represents the response of the Hessian determinant, which can be approximately defined as (4) [23, 24], where = 0.912 is a weight coefficient that is used to correct the error caused by approximation. The approximated *Dxx* , *Dyy* , and *Dxy* box filter kernels are depicted

in Fig. 3 [54], where white, gray, and black pixels refer to the weight values of {1,0 ，-2} for the *Dxx* and *Dyy* box filters and

{1,0，-1} for the *Dxy* box filter, respectively [55]. By the

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| concept of | the | integral | image, the calculation | of |
| *Dxx* or *Dyy* requires 8 memory accesses [54], while | | | | the |
| calculation of | *Dxy* | requires | 16 memory accesses. The 32 | |

memory accesses are marked with a dot in Fig. 3. The scale

is defined in the O-SURF method by an analogy with the linear scale space. The distances between the marked points increase with an increase of . However, the number of points to be accessed remains constant [56].

|  |  |  |
| --- | --- | --- |
| det( *A*) = *Dxx* *D* *yy* − 2 *Dxy* | 2 | (4) |

In the O-SURF method, the 9 9 box filter is defined for the first scale. However, Bay et al. did not specify exact values at

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

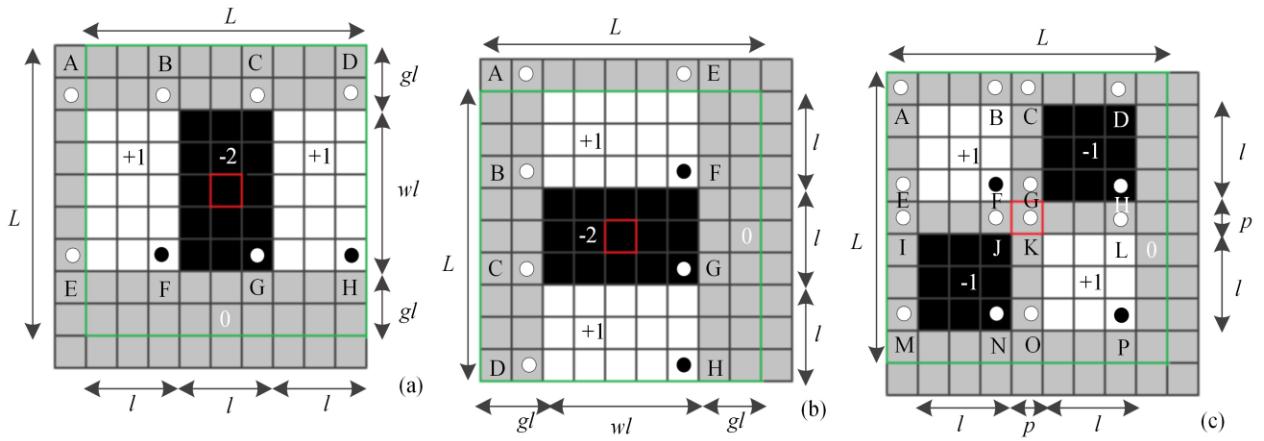
|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 4 |

the remaining scales [57, 58]. Hence, numerous parameters for the particular scale-levels (*i* , *j*) , *i* [1, *o*] and *j* [1, *v*] need to be

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| defined. | | | | Lobe *l* , | | which is one-third of *L* —the | | size | of the |  |
| *D* | *xx* | , | *D* | *yy* | ,and | *D* | box filters, is 2*o* *v* +1 . | The | scale |  |
|  |  |  |  | *xy* |  |  |  |  |

is (2 *o* *v* + 1) 1.23 = 0.4*l* . *wl* , which is the length of the

white area in the *Dxx* and *Dyy* box filters, is 2*l* +1 in [58] or 2*l* -1 in [59] or (3*l* +1)2 in MATLAB with OpenSURF by Chris Evans [60]. *gl* , which is the length of the side of the gray area in the *Dxx* and *Dyy* box filters, is (*l* −1)2 in [58] or (*l* +1)2 in [59] or (3*l* +1)4 in [60]. The size of the octave increases by 6 2*o*−1 pixels per interval. *p* is a constant of one pixel in the *Dxy* box filter. Table I shows the values of the box filters.



According to (2), the separable convolution response of the *Dxx* , *Dxy* and *Dyy* can be computed by (5) thru (7).

|  |  |  |  |
| --- | --- | --- | --- |
| *Dxx* | =(*A*+*F* −*B*−*E*)−2 (*B*+*G*−*C*−*F*) | (5) |  |
|  | +(*C*+*H* −*D*−*G*) |  |
|  |  |  |
| *Dyy* | =(*A*+*F* −*B*−*E*)−2 (*B*+*G*−*C*−*F*) | (6) |  |
|  | +(*C*+*H* −*D*−*G*) |  |
|  |  |  |
| *Dxy* | =(*A*+*F*−*B*−*E*)−(*C*+*H* −*G*−*D*)− | (7) |  |
|  | (*I* +*N* −*J* −*M*)+(*K* +*P*−*L*−*O*) |  |
|  |  |  |

Fig. 3. The 9×9 box filter that approximates the second-order Gaussian derivative in the *x* , *y* and *xy* directions to obtain the *Dxx* , *Dyy* , and *Dxy* box filters. (a)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Dxx* box filter. (b) *Dyy* | | box filter. (c) | *D*x*y* box filter. | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | TABLE I |  |  |  |  |  |
|  |  |  |  |  |  | BOX-SPACE SAMPLING VALUES | | |  |  |  |  |
|  | ***o*** | ***v*** | ***l*** | ***L*** | ***σ*** | **2*l+*1** | **(*l-*1)/2** | **2*l-*1** | **(*l*+1)/2** | **(3*l*+1)/2** | **(3*l*-1)/2** | |
| 1 | | 1 | 3 | 9 | 1.2 | 7 | 1 | 5 | 2 | 5 | 4 |  |
|  |  | 2 | 5 | 15 | 2.0 | 11 | 2 | 9 | 3 | 8 | 7 |  |
|  |  | 3 | 7 | 21 | 2.8 | 15 | 3 | 13 | 4 | 11 | 10 |  |
|  |  | 4 | 9 | 27 | 3.6 | 19 | 4 | 17 | 5 | 14 | 13 |  |
| 2 | | 1 | 5 | 15 | 2.0 | 11 | 2 | 9 | 3 | 8 | 7 |  |
|  |  | 2 | 9 | 27 | 3.6 | 19 | 4 | 17 | 5 | 14 | 13 |  |
|  |  | 3 | 13 | 39 | 5.2 | 27 | 6 | 25 | 7 | 20 | 19 |  |
|  |  | 4 | 17 | 51 | 6.8 | 35 | 8 | 33 | 9 | 26 | 25 |  |
| 3 | | 1 | 9 | 27 | 3.6 | 19 | 4 | 17 | 5 | 14 | 13 |  |
|  |  | 2 | 17 | 51 | 6.8 | 35 | 8 | 33 | 9 | 26 | 25 |  |
|  |  | 3 | 25 | 75 | 10 | 51 | 12 | 49 | 13 | 38 | 37 |  |
|  |  | 4 | 33 | 99 | 13.2 | 67 | 16 | 65 | 17 | 50 | 49 |  |
| 4 | | 1 | 17 | 51 | 6.8 | 35 | 8 | 33 | 9 | 26 | 25 |  |
|  |  | 2 | 33 | 99 | 13.2 | 67 | 16 | 65 | 17 | 50 | 49 |  |
|  |  | 3 | 49 | 147 | 19.6 | 99 | 24 | 97 | 25 | 74 | 73 |  |
|  |  | 4 | 65 | 195 | 26.0 | 131 | 32 | 129 | 33 | 98 | 97 |  |

*3) Non-maximum suppression*

To localize the interest point, the non-maximum suppression (NMS) is applied using the three adjacent scales. The NMS compares a determinant with its 8 direction neighbors in its native scale interval, and 9 direction neighbors in each of the intervals above and below for a total of 26 direction neighbors. Moreover, a threshold is employed to determine only the most distinctive image point as a candidate point [24].

*B. BRIEF Descriptor*

BRIEF is a descriptor that uses binary tests between two pixels in a smoothed image patch. More specifically, if *p* is a

smoothed image patch, the corresponding binary test is

|  |  |
| --- | --- |
| defined as (8) [26]. |  |
| 1 *if I* ( *p*, *x*) *I* ( *p*, *y*) |  |
| ( *p*; *x*, *y*) = | (8) |
| 0 *otherwise* |  |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | | | | | | | | 5 |  |
| where *p* ( *x*) | is the intensity of *p* at the point *x* . The descriptor | | | | | that only needs a 34-row memory foot to simultaneously | | |  |
| is defined as a vector of | | *nd* | binary tests [26]: | |  | calculate the response of two octaves box filters. The overflow | | |  |
|  | based on two's complement-coded arithmetic | | and rounding |  |
|  |  |  |  |  |  |  |
|  |  |  | *i* −1 |  |  | with error diffusion techniques were proposed by Belt [62], | | |  |
|  | *f nd* ( *p*)=2 ( *p*; *xi* , *yi* ) | | | | (9) |  |
|  | which can work on a face detector for a VGA resolution with a | | |  |
|  |  |  | 1 *i nd* |  |  |  |
| The length of *nd* is generally defined as 128-, 256-, and | | | | | | 16-bit vector. However, this approach has drawbacks, | | |  |
| including rounding errors and an additional constraint of a fixed | | |  |
| 512-bit vectors. The result of the experiment in [26] | | | | | | size for a box filter. LEE *et al.* [63] proposed a new structure for | | |  |
| demonstrated that the performance of 256-bit vector was | | | | | | memory size reduction which includes four types of image | | |  |
| similar to that of the 512-bit vector, while only marginally | | | | | | information: an integral image, a row integral image, a column | | |  |
| worse in other cases [48]. Due to the limited hardware | | | | | | integral image, and an input image. Using this method, the | | |  |
| resources, a 256-bit vector is employed in this paper. | | | | |  | integral image memory can be reduced to 42.6% for a 640×480 | | |  |
| *C. Hamming Distance Matching* | | | |  |  | 8-bit grayscale image. Ehsan *et al.* [64-66] proposed a parallel | | |  |
|  |  | recursive equation to compute the integral image. This method | | |  |
| In the BRIEF descriptor [26], the Hamming distance is used to | | | | | |  |
| not only substantially decreases the | operation | and memory |  |
| match. The Hamming distance can be efficiently calculated by | | | | | |  |
| requirements (by at least 44.44%) | but also | maintains the |  |
| the *XOR* | operation. | Considering | | the two | existing |  |
| accuracy. |  |  |  |
| descriptors *S*1 and *S*2 , the | | | corresponding | Hamming | distance |  |  |  |
| This paper focuses on reducing the size of the memory and | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  | 256 |  | parallelization computation. In [64], the maximum binary word | | |  |
| (256-bit) can be defined | | | as *Dhd* ( *S*1 , *S* 2 ) = (a *i* *bi* ) | | , where |  |
|  |  |  |  | *i* =1 |  | length of the integral image in the worst-case (WS) is stated as | | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *S*1 | = *a*1 *a*2 ,..., *a*256 , *S* 2 = *b*1*b*2 ,..., *b*256 , and the value of *ai* and *bi* | is | *ii*max | = (2 *Li* − 1) *W H* | (10) |  |
| 0 | or 1. The smaller is the value of *Dhd* , the higher is the | |  |
|  |  |  |  |

matching rate. Moreover, a threshold is used to check whether points truly correspond to each other. If the Hamming distance is less than a threshold, the feature points-pair are corresponding points; otherwise, they are nonmatching points [50].

1. OPTIMIZATION OF SURF DETECTOR

To ensure efficient implementation of the SURF detector in an FPGA, five approaches are applied to optimize the SURF detector. The first approach, which is word length reduction (WLR), is used to reduce the word length of the integral image without a loss of accuracy. The second approach, which is a memory-efficient parallel architecture (MEPA), is used to parallel compute the output of FIFO. The third approach, which consist of shift and subtraction strategies (SAS), is adopted to simplify the response of the Hessian determinant. The SAS transforms the floating-point operation into a shift and subtraction operation. The fourth approach utilizes sliding widow, is used to parallel compute the *Dxx* , *Dyy* and *Dxy* box

filters. The fifth approach is referred to a parallel multiscale-space. Five approaches are introduced in this section.

*A. Word Length Reduction (WLR)*

SURF is a detector and descriptor of local scale- and rotation-invariant image features. By using integral image for image convolution, SURF computes faster than other state-of-the-art algorithms but produces comparable or even better results by utilizing repeatability, distinctiveness, and robustness [59]. However, the word length of the integral image substantially impacts on the performance of designed hardware, especially for implementations that need to store the entire integral image on FPGA [55]. To solve this problem, Hsu et al. [61] presented a row-based stream processing (RBSP) method

where, *ii* is the word length of the integral image; *ii*max is the value of WS; *i* is the input image; *Li* is the word length per pixel of the input image. and *W* and *H* are the width and height, respectively, of the input image. According to [62], the number of bits *Lii* required for representing the WS integral

image value is (2 *Lii* − 1) (2 *Li* − 1) *W H* . The total memory in bytes required to store the integral image is (*W H* ) *Lii* 8 .

In general, the maximum width and height of the box filter are known, and the word length for the integral image using the exact method [66] with complement-coded arithmetic needs to satisfy:

|  |  |
| --- | --- |
| (2 *Lii* − 1) (2 *Li* − 1) *W*max *H*max | (11) |

where *W*max and *H* max are the maximum width and maximum height, respectively, of the box filter (i.e., *l* max *wl*max or *wl* max *l*maxin Fig. 3). For example, the input images comprise

8-bit gray data with resolution 512\*512 pixels. Table II lists the WLR values of the different sizes of the box filter with 4 octaves. The total memory in bytes is shown in Fig. 4. As shown in Table II and Fig. 4, the box filters in [58] and [59] have the same memory. The OpenSURF [60] has the smallest memory compared with that of [58] and [59]. However, the value of *l*2 is not an integer and is unsuitable for the FPGA.

Compared with the WS, WLR method substantially reduces the space of memory.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

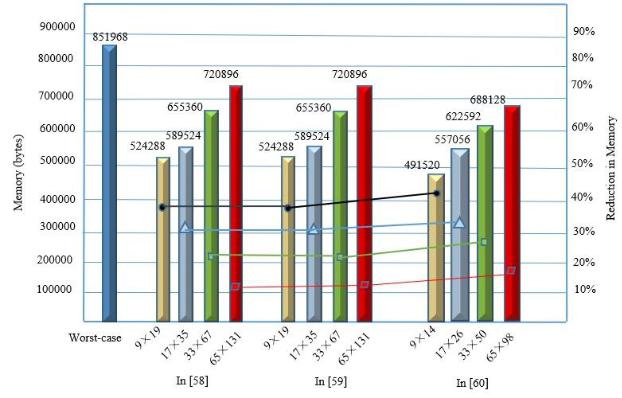
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 6 |

Table II

WORD LENGTH (bit) OF THE INTEGRAL IMAGE WITH THE MAXIMUM WIDTH AND HEIGHT OF THE BOX FILTER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Octave | WS *Lii* | *l* max*wl*max | *Lii* [58] | *l* max*wl*max | *Lii* [59] | *l* max*wl*max | *Lii* [60] |
| 1 | 26 | 9×19 | 16 | 9×17 | 16 | 9×14 | 15 |
| 2 |  | 17×35 | 18 | 17×33 | 18 | 17×26 | 17 |
| 3 |  | 33×67 | 20 | 33×65 | 20 | 33×50 | 19 |
| 4 |  | 65×131 | 22 | 65×129 | 22 | 65×98 | 21 |
|  |  |  |  |  |  |  |  |



*ii* ( *x* +2 *m* +1, *y* )= *ii* ( *x* +2 *m* −1, *y* )+ *S* ( *x* +2 *m*, *y*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | + *S* (*x* + 2*m* + | | | | 1, *y*) | |  |  |  | (16) |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | where *n* is the number of rows to be calculated (always a | | | | | | | | | | |  |
|  |  |  |  | multiple of 2), | *j* =0,...., *n* −1 | | | | , | *k* =0,...., *n* 2−1 | | | , | and |  |
|  |  |  |  | *m* =0,...., *n* 2−1 | . This | | set | |  | of | equations | | requires | |  |
|  |  |  |  | 2 *MN* + *MN* 2 addition operation for an input image with the | | | | | | | | | | |  |
|  |  |  |  | resolution *M N* pixels [66]. | | | | Compared with Viola-Jones | | | | | | |  |
|  |  |  |  | equation, the increase is not significant. | | | | | | |  |  |  |  |  |
|  |  |  |  | To ensure a trade-off between the computation time and | | | | | | | | | | |  |
|  |  |  |  | consumption memory, the 4-row parallel method is adopted in | | | | | | | | | | |  |
| Fig. 4. Comparison of memory between worst-case and maximum width | | |  | the integral image module. | | |  |  |  |  |  |  |  |  |  |
| and height of the box filter. | |  |  | *C. Shift and Subtraction (SAS)* | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| *B. Parallel Computation Integral Image* | | |  | In (4), the weight coefficient 2 | | | | | | = 0.831744 ( =0.912) is | | | | |  |
|  | derived to minimize the approximation error caused by the box | | | | | | | | | | |  |
| Equation (1) can be transformed into the pipeline recursive | | | |  |
| filters in the O-SURF. Hence, a floating-point architecture is | | | | | | | | | | |  |
| equation that was presented by Viola-Jones [67]: | | |  |  |
|  | required to compute det( *H* ) . OpenSURF does not use this | | | | | | | | | | |  |
|  | *S* ( *x* , *y* )= *i* ( *x* , *y* )+ *S* ( *x* , *y* −1) | |  |  |
|  | (12) | value, but instead applies 0.81 | | | | | ( = 0.9 ). However, | | | | | the |  |
|  | *ii* ( *x* , *y* )= *ii* ( *x* −1, *y* )+ *S* ( *x* , *y*) | |  | floating-point operation is more complex than that of a fixed | | | | | | | | | | |  |
|  | (13) | point. To overcome this problem, in Flex-SURF [55], the value | | | | | | | | | | |  |
| where *S* (*x* , *y*) | is the cumulative row sum value at the image | | | of 2 = 0.875 is employed. The same strategy is adopted to | | | | | | | | | | |  |
| simplify the processing in [35,48,50,69,70]. Equation (4) can | | | | | | | | | | |  |
| location ( *x* , *y*) . |  |  |  |  |
|  |  |  | be replaced by a subtraction and a shift operation [69]. | | | | | | | | |  |  |  |
| In (1), *M* 2 *N* 2 | 4 adders are used to | compute the integral | |  |  |  |
| det(*H* |  | )=*DD* | | | − 0.875(*D* )2 | | | |  |  |  |
| image for an image with resolution | | *M N* pixels | [68]. | *approx* |  |  |  |
|  |  | *xx* | *xy* |  |  | *xy* |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Apparently, Equation (1) is not suitable for a medium- or | | | |  | = *Dxx* *Dxy* − ( *Dxy* | | | | | | 2 − *Dxy* | 2 /8) |  | (17) |  |
| high-resolution image. In the Viola-Jones parallel recursive (12) | | | |  | =*D D* −*D*2 | | | | | | + ( *D* | 23) |  |  |  |
| thru (13), the number of additions is 2*MN* . However, the | | | |  |  | *xx* | | *xy* |  | *xy* | *xy* | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Viola-Jones method has time delay drawbacks. To accelerate | | | | Fig. 5 shows the number of operations which are employed | | | | | | | | | | |  |
| the processing of an integral image. Ehsan et al. [66] proposed | | | | to calculate the integral image using the SAS and the O-SURF. | | | | | | | | | | |  |
| an *n* stages of a pipelined system that processes *n* rows of an | | | | As observed in Fig. 5, the SAS highly requires a larger number | | | | | | | | | | |  |
| input image in parallel, providing *n* integral image value per | | | | of shift operations than O-SURF. However, the shift operation | | | | | | | | | | |  |
| clock cycles without delay when the pipeline is full. This | | | | consumes one clock period in the FPGA architecture. | | | | | | | | | | |  |
| method can be mathematically defined as: | | |  | Compared with the O-SURF, addition/subtraction (add-sub), | | | | | | | | | | |  |
| *S* ( *x* + *j* , *y* )= *ii* ( *x* + *j* , *y* )+ *S* ( *x* + *j* , *y* −1) | | | (14) | multiplication and division are reduced by 4.44%, 13.33%, and | | | | | | | | | | |  |
| 33.33%, respectively. | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| For odd rows: |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *ii* ( *x* +2 *k* , *y* )= *ii* ( *x* +2 *k* −1, *y* )+ *S* ( *x* +2 *k* , *y*) | | | (15) |  |  |  |  |  |  |  |  |  |  |  |  |
| For even rows: | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 7 |

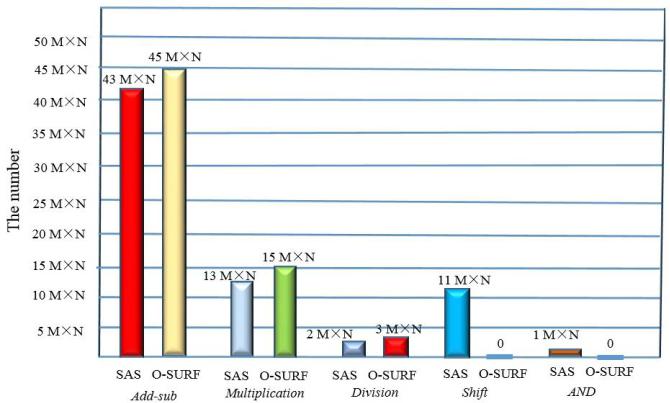


Fig. 5. Comparison between the SAS and the O-SURF for an image with the resolution *M N* pixels.

*D.* *Sliding Window*

The sliding window technique has been indicated to be a good choice for parallel computer data [30, 49, 71, 72], which includes four portions: the stream of an input image, a buffer, slice registers, and a function module. The described fabric is depicted in Fig. 6. The input data stream is buffered in a custom pipeline structure and organized in pixel rows. The buffer is implemented by a combination of block random-access memory first in, first out (Block-RAM FIFO) and slice registers (SRs). The SR sections enable access to all pixels in the respective pipeline elements compared with a window. With each incoming pixel, data are shifted by one pixel, which causes the window to virtually slide forward. Hence, while the input image is streamed into the pipeline, the window moves grid by grid from the top left (TL) to the bottom right (BR) in the integral image of the original image and reads the corresponding data from the slice register. A function module, which is determined for a certain window operation, interconnects to a subset of pixel registers to simultaneously fetch all data required for one calculation. With each pixel shifted into the structure, the function module calculates a new result and produces an output data stream [54]. The module has three types of sliding windows (*N*×*N*) in the total system, where *N* is equal to 52, 5, and 35 in the Hessian response, non-maxsuppression and BRIEF descriptor, respectively.

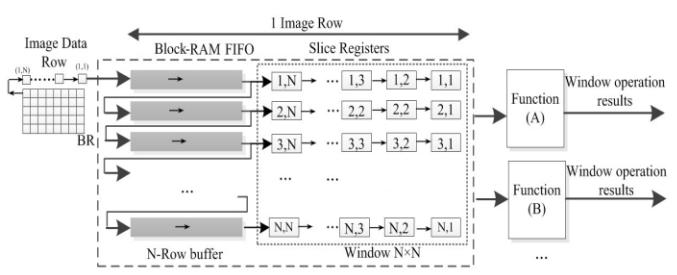


Fig. 6. Architecture of the sliding window.

*E.* *Parallel Multiscale-Space for Hessian Determinant*

The larger is the number octaves, the larger is the number hardware resources that will be consumed [73]. Avoiding consumes more resources in the FPGA, and the parallel multiscale space architecture is simultaneously designed to compute the Hessian determinant. The recommended two octaves and six scales [24] are implemented to extract feature

points, which corresponds to the scales {9, 15, 21, 27} and {15, 27, 39, 51} respectively.

The interpolation step in the Hessian determinant of the O-SURF is computationally expensive because it requires the calculation of the first- and, second-order derivatives of the Hessian matrices and their inverses. Two box filters with sizes

1. (*l*=11, *L*=33, 2*l*-1=21, (2*l*+1)/2=7) and 45 (*l*=15, *L*=45, 2*l*-1=29, (2*l*+1)/2=8) are added to calculate the Hessian determinants to create a scale-space with higher granularity and remove the interpolation step without sacrificing the accuracy [74]. A total of 8 scales {9, 15, 21, 27, 33, 39, 45, 51} are used to compute the Hessian determinants.

To efficiently calculate the Hessian determinant, multiple integral images need to be accessed by row-based stream processing (RBSP) [61] in parallel to perform the separable convolution of the integral image with 24 box filters (8×3). The RBSP cores are visualized in Fig. 7. Fig. 7(a) lists the 10-line buffers for the box filter of size 15×15. Fig. 7(b) reveals that the memory foot of 32 (8+8+16) points of the box filters in Fig. 7(a), where *Wx* , *Wy* and *W* represent the corresponding points

in the *Dxx* , *Dyy* and *Dxy* box filters. Fig. 7(c) shows the memory

foot of 24 box filters with the size {9, 15, 21, 27, 33, 39, 45, 51}.

The 32 sampled points are a separable convolution with the

15 15 box filter as follows: *L*0 thru *L*15 data are parallel input to the r-line buffer cores. Sixteen registers ( *R*0 thru *R*15 ) are

used to store one-line data, and 32 points can be selected to calculate the determinant of the Fast-Hessian. The box filter responses are given as

*Dxx* =( *L*3\_ *R*15+ *L*12\_ *R*10− *L*3\_ *R*10− *L*12\_ *R*15)

−2 (*L*3\_*R*10+*L*12 \_*R*5−*L*3\_*R*5−*L*12 \_*R*10)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| +(*L*3 \_*R*5 +*L*12 | \_ *R*0 | −*L*3\_*R*0 | − *L*12 | \_ *R*5) | (18) |  |
| = (*L*3 \_ *R*15 − *L*3 \_ *R*10)−(*L*12 \_ *R*15 − *L*12 \_ *R*10) | | | | |  |
|  |  |

* 2 (*L*3 \_ *R*10 − *L*3 \_ *R*5)+2 (*L*12 \_ *R*10 − *L*12 \_ *R*5) + (*L*3 \_ *R*5 − *L*3 \_ *R*0）−（*L*12 \_ *R*5 − *L*12 \_ *R*0 )

*D yy* =( *L*0\_ *R*12+ *L*5\_ *R*3− *L*0\_ *R*3− *L*5\_ *R*12)

* 2 (*L*5\_*R*12 +*L*10 \_*R*3− *L*5\_*R*3−*L*10 \_*R*12)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| + (*L*10 | \_*R*12 +*L*15\_*R*3−*L*10 | \_*R*3−*L*15 | \_ *R*12) | (19) |  |
| =(*L*0 \_R12− *L*0 \_ *R*3) − (*L*5 \_ *R*12 − *L*5 \_ *R*3) | | | |  |
|  |  |

* 2 (*L*5 \_*R*12 −*L*5 \_*R*3)+2 (*L*10 \_*R*12 −*L*10 \_*R*3) +(*L*10 \_ *R*12 − *L*10 \_ *R*3 ) − (*L*15 \_ *R*12 − *L*15 \_ *R*3 )

|  |  |
| --- | --- |
| *Dxy* =( *L*2\_ *R*13− *L*2\_ *R*2)−( *L*7\_ *R*13− *L*7\_ *R*8) |  |
| − ( *L*2 \_ *R*6 − *L*2 \_ *R*2 ) + ( *L*7 \_ *v*7 − *L*7 \_ *R*2 ) | (20) |

* (*L*8 \_ *R*13 − *L*8 \_ *R*8 )+(*L*13 \_ *R*13 − *L*13 \_ *R*8 )

Because the values of *R*0thru *R*15are integral, then the

multiply operation can be transformed into a shift operation. Equations (18) thru (20) are decomposed into the vertical and horizontal convolution, which are implemented by addition/subtraction and a shift operation. The proposed

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

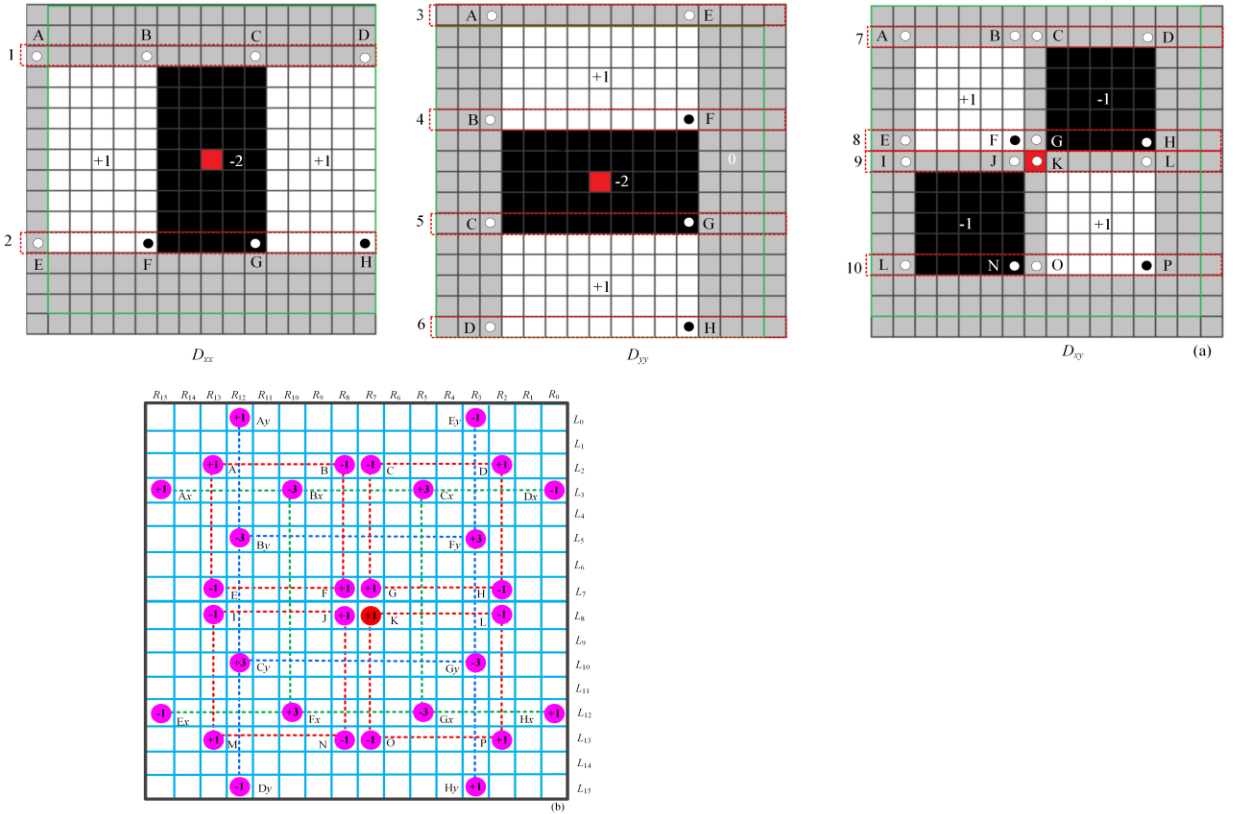
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 8 |

separable convolution method is simpler than proposed by Čížek [75]. The proposed separable convolution utilization is only *O* ( *n*) and is not *O* ( *n*2 ) for parallel implementation of the

full sliding window [75].

In Fig. 7(c), the color dots are the selected positions for the convolution operation. The sliding window shifts from the left



to the right and from the top to the bottom of the image. After scanning through the whole image, the Hessian determinants are simultaneously computed at the same period clock.

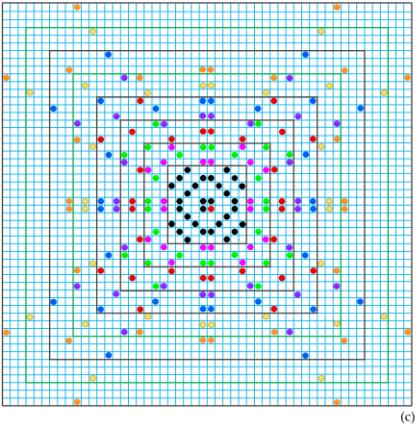
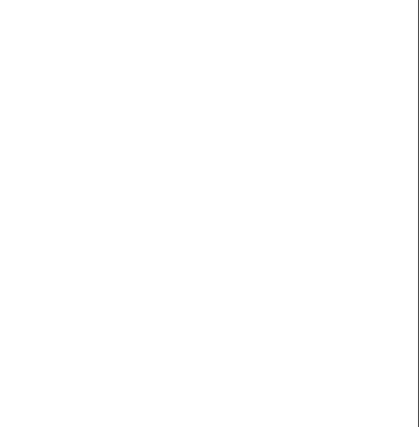


Fig. 7. Parallel Multi-Scale Space Hessian Detector. (a) ten-line buffers for the box filter of size 15×15; (b) sampled points of the 15 15 box filter; (c) pixel-access window is required for parallel convolution with the 24 box filters.

IV. HARDWARE IMPLEMENTATION

*A.* *Total System Architecture*

Considering the memory-space, power, and real-time constraints of embedded systems, the FPGA is selected to ensure the system performing in real-time. The proposed total hardware architecture, which is shown in Fig. 8, contains memory controller module, integral image generation (IIG) module, SURF detector module, BRIEF descriptor module, and BRIEF matching module.

(1) Memory controller module

To drive onboard, DDR3 and Xilinx IP memory interface generator (MIG) are chosen to create a logical connection with DDR3 [32].

(2) IIG module

An integral image is a novel method for improving the performance of the SURF detector. The WLR algorithm and 4-row parallel method are adopted to optimize the integral image. The IIG module transforms the integral image to the SURF feature detector module, memory controller module and BRIEF descriptor module. Therefore, the IIG module is separated from the SURF feature detector module.

(3) SURF detector module

The SURF detector extracts the local maxima Fast-Hessian determinant as candidate points on the multiscale and then locates the corresponding index and scale. The FPGA-based SURF feature detection architecture is divided into two submodules: Fast-Hessian response generation and the location of an interest point. The location of interest point is divided into three steps: non-maximal suppression, threshold, and interpolation. To solve these problems, a parallel architecture for the modified SURF algorithm is proposed. A sliding window buffer is used to store the shifted pixels of an integral image for each clock. The buffer is shared with the Hessian determinant. The SAS algorithm and parallel multi-scale space are used to implement the Hessian determinant. An additional 33 and 45 scales are used to replace the step of interpolation, without sacrificing the accuracy [75].

(4) BRIEF descriptor and matching module

A 256-bit BRIEF descriptor and matching require a low hardware cost. To reduce the complexity of the BRIEF descriptor, optimized parallel adder trees and parallel comparators are employed for the BRIEF descriptor and matching.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 9 |

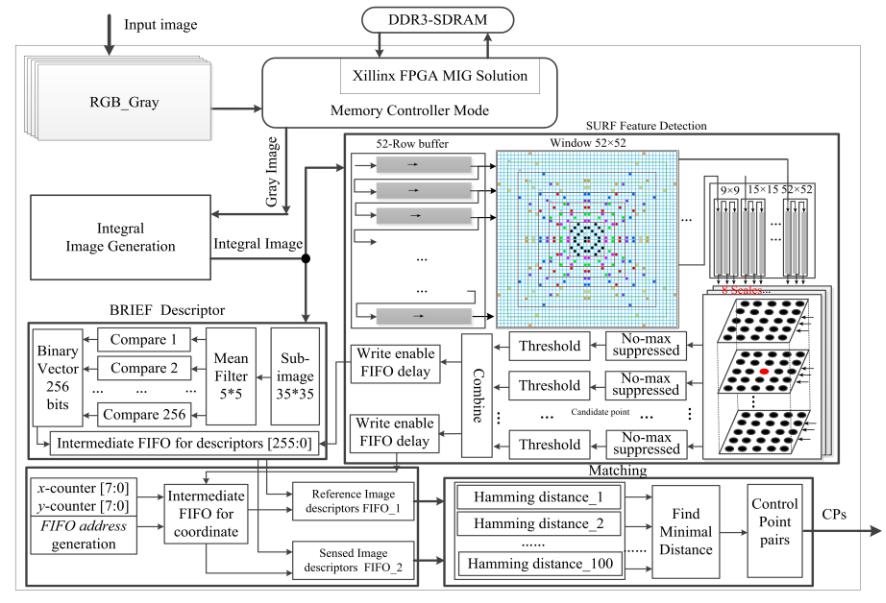


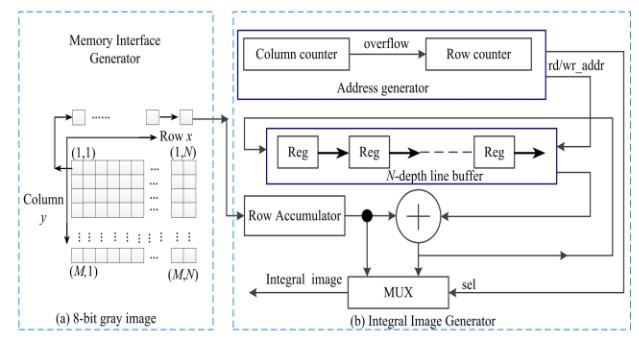
Fig. 8. The proposed hardware architecture.

*B.* *Integral Image Generator* （*IIG*）

IIG module generates the integral image from the incoming 8-bit grayscale image via the memory interface generator (MIG)

1. to communicate with off-chip on-board DDR3 SDRAM (refer to Fig. 9(a)), which are stored by the row sequence.

The hardware architecture of the integral image (refer to Fig. 9(b)) consists of an address generator, row accumulator, multiplexer, and adder. The address generator module is designed to generate the read address (rd\_addr) and write address (wr\_addr) via the column and the row counter. The value of the row counter generates the selector (sel) signal of the multiplexer (MUX).



construct a sliding window for the Hessian response, a FIFO architecture includes 52-line buffers and 52×52 silence registers are used. The sliding window provides parallel access to the integral image that is required for the second-order Gaussian derivatives *Dxx* , *Dyy* and *Dxy* using (18) thru (20).

The 256 pixels (due to overlap, 220 pixels exist) enable access to the response of the 8 box filters.

After *Dxx* , *Dyy* , and *Dxy* are obtained, the Hessian response

can be calculated in the three pipelines by using (17), as shown in Fig. 11. 8 Hessian determinants are concurrently calculated and then parallel output to the non-maximal suppression module.

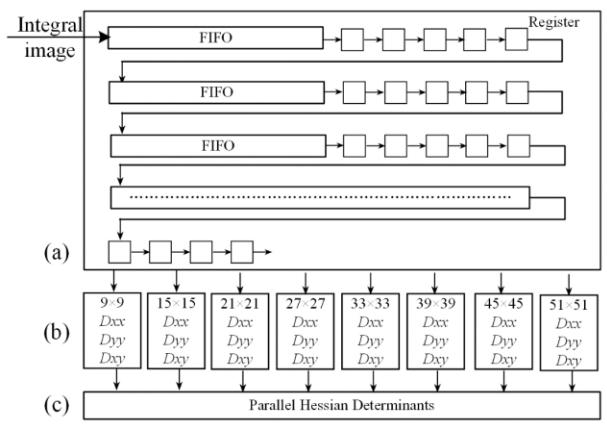


Fig. 9. Hardware architecture for the integral image generator. (a) Memory interface generator; (b) Integral image generator.

*C.* *SURF Detector Implementation*

*1) Fast-Hessian Responses Implementation*

After calculating the integral image in the IIG module, the integral image is sent to 52-line buffer [76] and 52×52 silence registers in the sliding window for the separable convolution. The detailed design of the multiscale-space Hessian detector is visualized in Fig. 10. This architecture includes the sliding window, parallel implementation second-order Gaussian derivatives, and parallel calculation Hessian determinants. To

Fig. 10. Parallel multiscale-space for fast Hessian response. (a) Sliding window. (b) Parallel implementation *Dxx* , *Dyy* , and *Dxy* . (c) Parallel calculation

Hessian determinant.

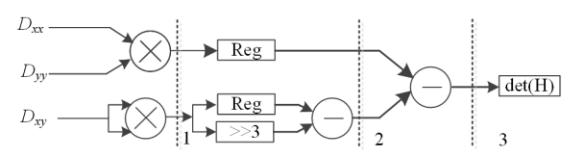
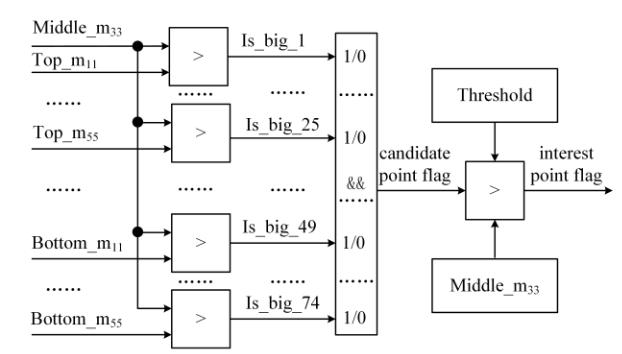


Fig. 11. Architecture for the pipelined Hessian determinant calculation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 10 |



*2) Non-Maximal Suppression Implementation*

Non-maximal suppression (NMS) module selects the local maximal Hessian determinant as a candidate point. A 5 5 sliding window is chosen in the NMS module, as shown in Fig. 12 [74]. 8 multi-scales can be divided into 6 multiscale-spaces to concurrently obtain the local maximal of interest point. The 5 5 points of each matrix can be presented as Top\_m*i* , *j* , Middle\_m*i* , *j* and Bottom\_m*i* , *j* (*i* = 1, 2, 3, 4, 5;

1. = 1, 2, 3, 4, 5) .The center point Middle\_m3,3 is compared with

its 74 neighbor points (24 neighbors on the same scale and 25

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| neighbors in the consecutive scales above and below) in | | | | | Fig.13. Architecture of location maximum across the 74 neighbor points. | | | | | | | |  |
| parallel. If the result of the *AND* operation is true, the center | | | | | *D.* | *BRIEF Descriptor Implementation* | | | | |  |  |  |
| point is regarded a candidate point [50], as shown in Fig. 13. | | | | |  |  |  |
| The BRIEF-32 [26] algorithm is adopted | | | | | | to generate | |  |
| The candidate point is fed to the user-defined threshold module. | | | | |  |
| descriptors. As shown in Fig. 14, the BRIEF-32 structure | | | | | | | |  |
| Only the candidate point which is great than the threshold can | | | | |  |
| includes two modules: image buffer and point pair comparator. | | | | | | | |  |
| be considered as a feature point, and the 1-bit interest point flag | | | | |  |
| In | the image | buffer | module, a | | 35×35 sub-window | | was |  |
| is set to *true* [72,74]. |  |  |  |  |  |
|  |  |  |  | presented by Huang [50]. Calonder et al. [26] provided the time | | | | | | | |  |
| The 8-scale Hessian determinants are parallelized to compute | | | | |  |
| required for Gaussian smoothing, a simple box filter, and a box | | | | | | | |  |
| it synchronously. The | search for a | local | maximum | in the |  |
| filter using the integral image, the latter was considerably faster. | | | | | | | |  |
| scale-space domain is | also pipelined | and | parallelized | using |  |
| Furthermore, no matching performance loss occurred. In this | | | | | | | |  |
| register FIFOs. Only the determinants that exceed a certain | | | | |  |
| paper, a 5×5 box filter using integral images is used to smooth | | | | | | | |  |
| threshold are saved, while the other determinants, which are set | | | | |  |
| the | original | image. | The | point | comparator | module | has |  |
| to zeros points and not local maxima in the three-dimensional, | | | | |  |
| 256-binary (32-byte) patch tests, which are related to blue lines | | | | | | | |  |
| scale-space domain, are suppressed (=0). This method ensures | | | | |  |
| that | are sampled from | | | an isotropic (0, *S* 2 | | 25) Gaussian | |  |
| that features do not overlap and are evenly spread over the input | | | | |  |
| image [74]. Every local maximum is compared with a | | | | | distribution. A total of 256 patch-points *p* ( *ri* , *c* *j* ) (*i* = 1,..., 256; | | | | | | | |  |
| user-defined threshold value. The threshold controls the total | | | | | *j* =1,..., 256)are parallel compared with the corresponding | | | | | | | |  |
| sensitivity of the detector by fine-tuning the number of interest | | | | |  |
| points using (8) in the same cycle, and the 1-bit result with a | | | | | | | |  |
| point that populate the image [58]. Considering that the features | | | | |  |
| comparison of | | 1or 0 is stored in a 256-bit descriptor register in | | | | | |  |
| in the same octave will generate the same descriptor if they | | | | |  |
|  |  |  |  |  |  |  |  |  |



|  |  |  |
| --- | --- | --- |
| have the same coordinates but a different scale, we only store | sequence. |  |
|  |  |
| one of them if more than one feature simultaneously exists [77]. |  |  |

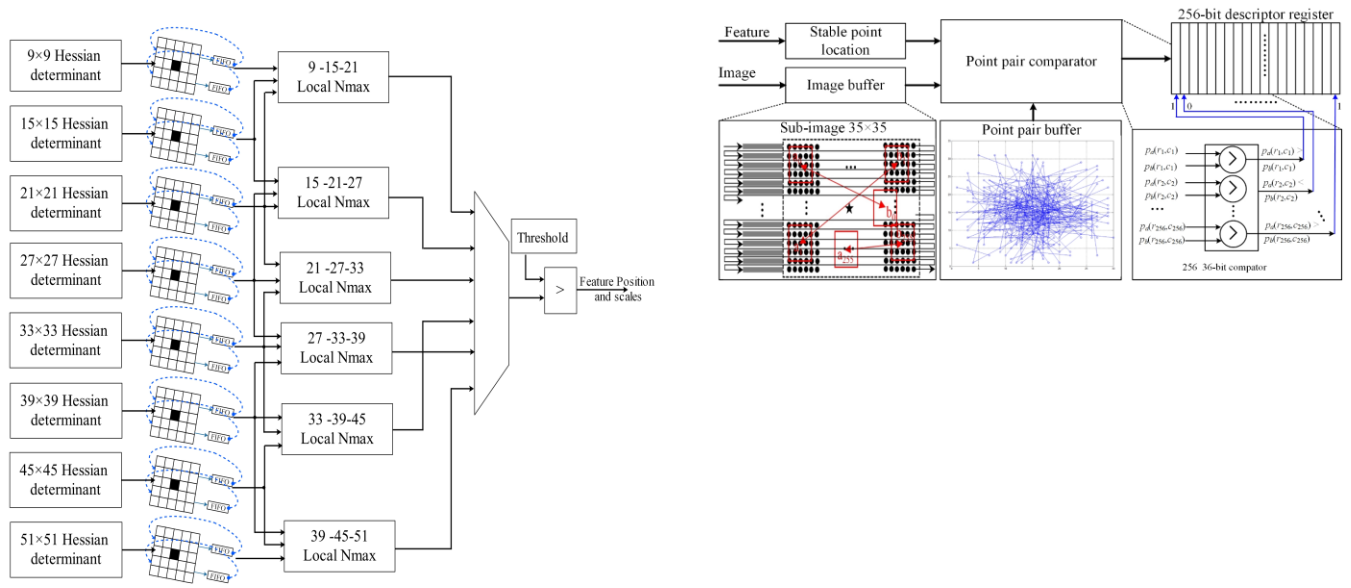


Fig. 14. Exemplary 256 patch-pairs for BRIEF descriptor.

Fig. 12. Architecture of non-maximal suppression module.

*E. BRIEF Matching Implementation*

The Hamming distance is used to match the 256-bit descriptor in the BRIEF matching. The BRIEF descriptor is robust to illumination changes and small rotations [26], which renders it an excellent candidate point for the georeferencing. The BRIEF matching module consists of computing the Hamming distance module and finding the minimal Hamming distance module. The reference image descriptors are stored in FIFO1, and the sensed image descriptors are stored in FIFO2. To ensure a trade-off the speed time and resource. The number of descriptor points determines the accuracy of the matching. However, the larger is the number of descriptor points, the larger is the number resources that are consumed. For example,

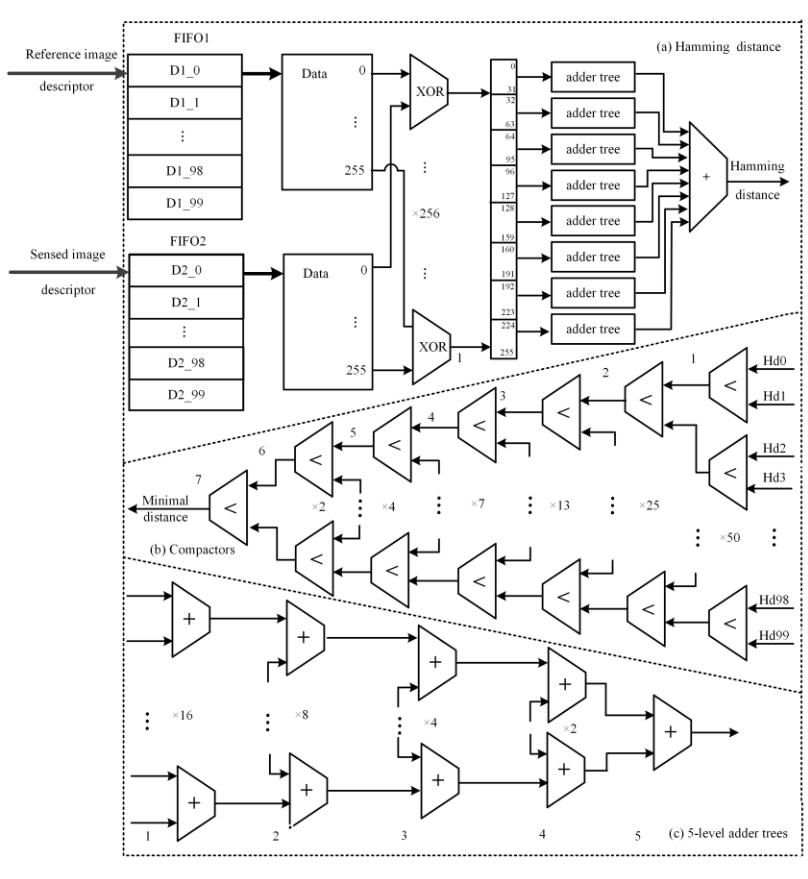
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 11 |

100 pair-descriptors are used to parallel implement the Hamming distance in Fig. 15 [50]. In the Hamming distance module (refer to Fig. 15(a)), the Hamming distance is computed using 256 *XOR* gates, and the results are stored in a 256-bit register. Improved parallel pipelined adder trees [77] are used to compute the number of '1s' in the 256-bit register. The 256-bit result of the *XOR* is divided into 8 32-bit registers, which are paralleled to calculate the Hamming distance by 5-level pipeline adder trees (refer to Fig. 15(c)). However, 9-level pipeline adder trees were employed in [77].

The finding minimal Hamming distance module begins to work when the 100 Hamming distances are received. The minimum Hamming distance means the best matching. The



corresponding matching feature point is defined as the GCP. The architecture of the finding minimal Hamming distance module is shown in Fig. 15(b), and an improved compactor module is used to compare the Hamming distance with 7 levels of pipeline. Three compactor modules are reduced compared with [50].

Based on the matching algorithms, the best matching feature points are the GCPs. The coordinates of GCPs are the scanning coordinates. However, in the georeferencing method, the geodetic coordinates are used in the projection transformation equation. Thus, the scanning coordinates must be transformed into geodetic coordinates [1].

Fig. 15. Structure of matching coprocessor. (a) Computation Hamming distance. (b) Locating minimal Hamming distance. (c) 5-level adder trees.

* 1. EXPERIMENT

1. *Hardware Environment and Data Set*

The proposed system is implemented in a signal Xilinx XC7VX980T FPGA that has 612,000 logic cells, 1,224,000 Flip-Flops, 1,500 kB Block RAM, and 3,600 DSP slices. The development kit Vivado (14.2 version) is used to design the hardware of the system in Verilog HDL, and the simulation tool is Vivado simulator. The first data sets are obtained from [50] (Fig. 17 (a) and (b)), the second data sets are downloaded from the BIGMAP software (Fig. 17(c) and (d)). The image resolution of 512×512, and a 100MHz working frequency are assumed. Additionally, the results that are generated by the implemented FPGA are compared with those of the OpenCV library, which is written by Chris Evans in the MATLAB. As

expected, the results are identical.

B. Interest point Analysis

The number of interest points is affected by some parameters such as octave, scale, resampling, size of the mon-maximal matrix, and the threshold [24, 50, 80]. Table III shows the number of interest point with the different thresholds. As seen from Table III, the variation in the threshold has a significant effect on the interest point distribution, which is consistent with [72,74,80].

Fig. 16 shows the results of the different textures pair-points matching. In Fig. 16(a), the number of matching and mismatching points is 79 and 21, respectively. When the image pairs are covered with the high-rise building in Fig. 16(b), the matching points and mismatching points are 83 and 17

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 12 |



respectively. There are 700 matching points and 100 mismatching points in Fig. 16 (c). And there are 90 matching points and 130 mismatching points in the Fig. 16(d). The results indicated that the uniform distribution of the matching points and the matching rate are affected by the number of matching points and the textures of the object. Additionally, some errors occur in the matched point pairs; however, these points can be eliminated by using robust fitting methods, such as RANSAC [15, 27, 42, 81] or a combined algorithm of slope-based rejection (SR) and correlation-coefficient-based rejection (CCR) [48].

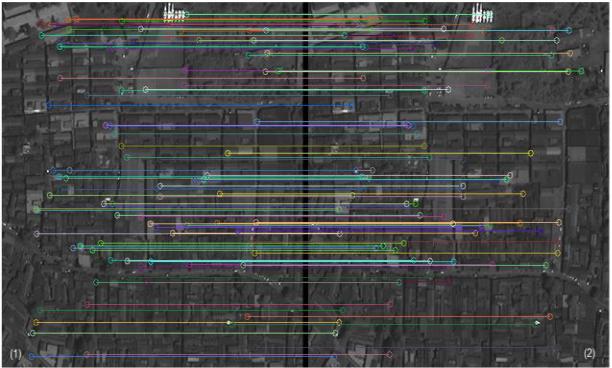
TABLE III

NUMBER OF FEATURE POINTS WITH DIFFERENT THRESHOLDs

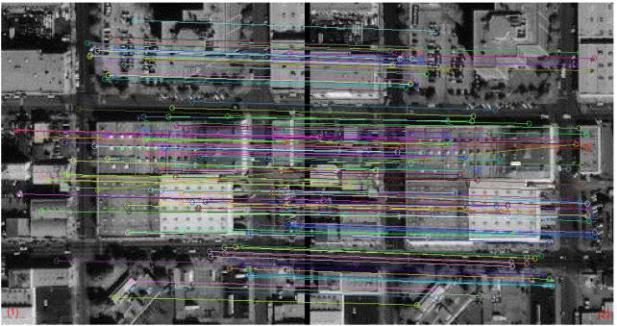
(THs).



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Image pairs | No TH | TH=1000 | TH=8000 | TH=15400 | |  |
|  |  |  |  |  |  |  |  |
|  | Bungalows | 6666/6526 | 1467/1452 | 125/118 | 68/63 |  |  |
|  | 1/2 |  |  |
|  |  |  |  |  |  |  |
|  | High-rise | 10110/10548 | 2309/2277 | 1169/1125 | 703/576 |  |  |
|  | building 1/2 |  |  |
|  |  |  |  |  |  |  |
|  | expressway | 3898/3928 | 2931/2918 | 12701220 | 718/683 |  |  |
|  | 1/2 |  |  |
|  |  |  |  |  |  |  |
|  | Bare soil 1/2 | 3349/3349 | 491/491 | 32/32 | 26/26 |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

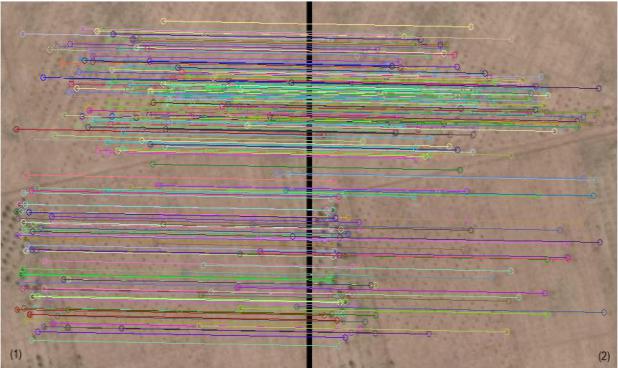


(a)



(b)

(c)



(d)

Fig. 16. Matched remote sensing image pairs. (a) Matched bungalows image pairs with the best 100 matching pair-points; (b) Matched high-rise building pairs with the best 100 matching pair-points; (c) Matched expressway pairs with the best 800 matching pair-points; (d) Matched bare soil pairs with the best 220 matching pair-points.

*C.* *Accuracy Analysis*

The *recall* versus *1−precision* curves [17] describe useful characteristics of a feature's performance, and are widely used as standard criterion, which is defined as:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *recall* = | # *correct matches* |  | (21) |  |
|  | # *correspondences* | |  |
|  |  |  |  |
| 1− *precision* = | # *false matches* | | | (22) |  |
|  |  |  |  |
| # *correct matches* + # *false matches* | | |  |
|  |  |  |

where *recall* is the ratio of the number of correctly matched points relative over the number of corresponding matched points. 1*−precision* is defined as the ratio of the number of false matches to the total number of matches, which includes the false matches and the correct matches. The curves are generated below a threshold *t*, which determines whether two descriptors are matched, if *recall* is increasing and 1*−precision* is equal to 0, it means that the point pairs are all correctly matched without any mismatching; if *recall* is static and 1*−precision* is increasing, it means that the number of falsely point pairs is increasing, while the correctly point pairs remain unchanged [40].

SURF+BRIEF method has a better matching performance in different land coverages than that of SURF in the software (OpenCV2.4.9 + Microsoft Visual Studio 2015) [50]. In this paper, the performance evaluation of the FPGA-based

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 13 |

implementation is also conducted and compared with that of OpenSURF in MATLAB [60]. The relationship between the two images can be described by the Homography matrix [40].

The Homography matrices of the four image pairs are calculated in advance by MATLAB on a PC. The results are listed as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.024682517497177 | 0.018725015152393 | -5.404435483561459 | |  |
|  |  |  |  |  |
| *Hbungalows* =0.015206213894699 | 1.014273691140137 | -2.552171349203126 | | (23) |
| 0.000024222830514 | 0.000023299587958 | 1.000000000000000 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.099260397052538 | 0.002259608451860 | -7.957861186118295 | |  |
|  |  |  |  |  |
| *Hhigh* −*rise building* =0.090326101809412 | 0.955409181508876 | -4.421034816054379 | | (24) |
| 0.000271661969942 | -0.000085315081328 | 1.000000000000000 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.000297662301777 | | 0.025695798331601 | -66.220856181085963 | |  |
|  |  |  |  |  |  |
| *Hexpressway* = | -0.032698271963169 | 1.023565673743905 | 8.462326102920549 |  | (25) |
|  | -0.000064630077908 | 0.000065894730718 | 1.000000000000000 |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.061661003420602 | | -0.064774241169707 | -22.139327368389228 | |  |
|  |  |  |  |  |  |
| *Hbare soil* = | 0.030845167584049 | 0.861666704860821 | 6.043495802072064 |  | (26) |
|  | 0.000065697594479 | -0.000382940095883 | 1.000000000000000 |  |  |
|  |  |  |  |  |  |

Inspired by [40], the number of detected points in each image is also defined as approximate 100 and there are 100 point-pairs output by the PC. The higher value of *recall* and the lower value of 1*−precision* mean that the better performance of matching. When 1*−precision* = 0 (i.e., the number of falsely point pairs is 0), the value of *recall* is equal to “m”, which means “100×m” correctly point pairs (the maximum number of point pairs is 100). While 1*−precision* stops at value “n”, the number of falsely point pairs is “100×n”. The curves of *recall* versus 1*−precision* of FPGA and PC implemented are shown in Fig. 17.

In the bungalow image pairs (Fig. 17a), the curve of FPGA implemented is slightly lower than that of PC-based. In other words, the performance of the PC-based is slightly better than that of the FPGA’s. In addition, when 1*−precision*=0, the *recall* of FPGA-based and PC-based are approximately 0.79 and 0.81, respectively. Finally, the 1*−precision* of FPGA-based and PC-based stop at approximately 0.20 and 0.14, respectively.

In the high-rise building image pairs (Fig. 17b), the curve of PC-based is slightly higher than that of FPGA-based. When 1*−precision*=0, the *recall* of FPGA-based and PC-based are approximately 0.82 and 0.85, respectively. Finally, the 1*−precision* of FPGA-based and PC-based stop at approximately 0.15 and 0.10, respectively.

The expressway texture in Fig. 17c, the PC-based curve is slightly higher than that of FPGA-based. When 1*−precision*=0, the *recall* of FPGA-based and PC-based are approximately 0.70 and 0.71, respectively. Finally, the 1*−precision* of FPGA-based and PC-based stop at approximately 0.23 and 0.20, respectively.

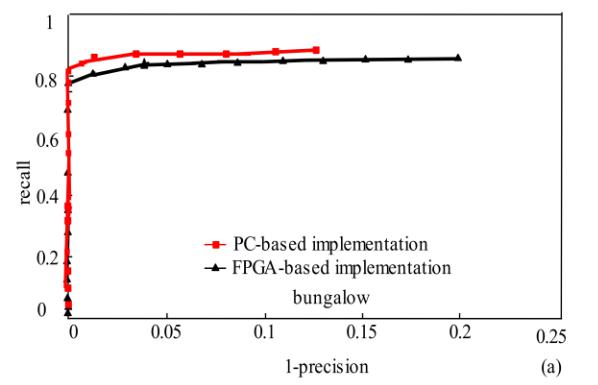
In the bare soil texture image pairs (Fig. 17d), the FPGA-based curve is slightly higher than that of PC-based. When 1*−precision*=0, the *recall* of FPGA-based and PC-based are approximately 0.30 and 0.29, respectively. Finally, the

1*−precision* of FPGA-based and PC-based stop at approximately 0.41 and 0.44, respectively.

The combined SURF+BRIEF method was also presented in [50,72], and the same criterion *recall* versus *1−precision* curves is used to evaluate the matching performance. Experiments indicated that the combined SURF+BRIEF method has a similar performance with [40, 50, 72]. Hence, the final computation accuracy is acceptable.

As shown in Fig. 17, the performance of the FPGA-based implementation is slightly worse than that of the PC-based implementation. The reasons are listed as follows [50,73]:

1. Error is inevitable when the floating-point data are approximated with fixed-point data, and an error analysis is conducted [72]. To save logic resource, several fixed-point approximations are implemented to save resources and fit the entire architecture on a single Xilinx chip. The Hessian determinant is approximated using only shifting and the adding operation [35,48,50,69,70];
2. Only two octaves are implemented to detect feature points, which will inevitably cause performance degradation.

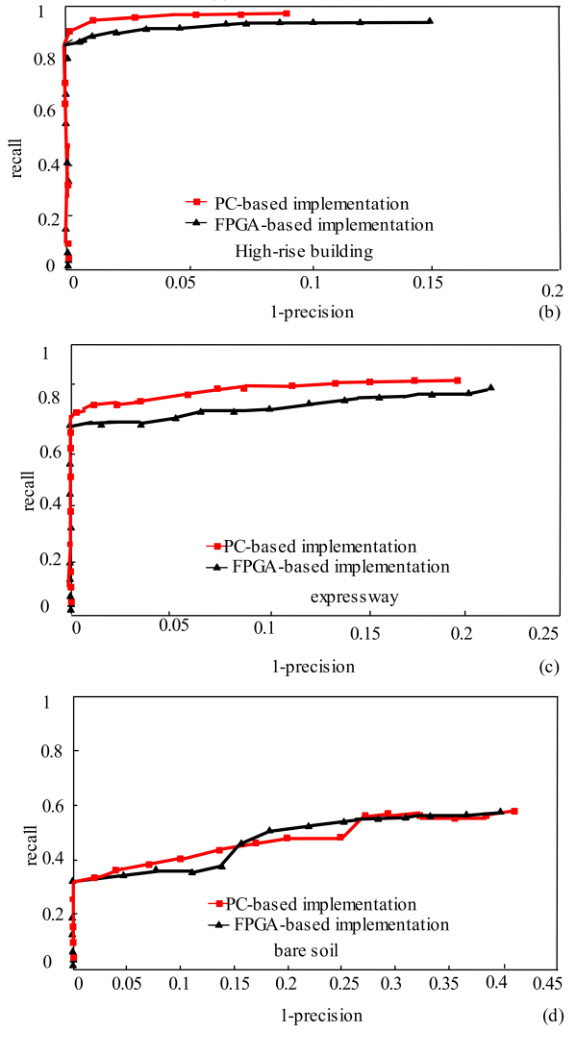


This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 14 |

Fig. 17. *Recall* versus 1*−precision* of four RS image pairs. (a)



Bungalows; (b) Rise-high building; (c) Expressway; and (d) Bare soil.

*D. Performance of FPGA Analysis*

*1) FPGA Resources Utilization*

A total of 298,864 (48.8%) slice LUTs, 267,095 (21.82%) slice FFs, 144 DSP (4%) and 11Kb (0.73%) memories that are used to implement the proposed architecture. Two octaves and 8 scales (6 O-SURF scales and two extra scales (33 and 45)) are used to build the Hessian response. The SAS, sliding window, parallel multiscale-space, and parallel pipelined add-trees are used to optimize the SURF detector and BRIEF descriptor. Compared with [27, 53, 72], the BRAMs resource is significantly reduced but the slice logical resource substantially increases compared with [27, 32, 53, 72, 74, 78]. In [50,72], The same algorithm was proposed, but only six scales were used to compute the Hessian detection, and the interpolation step was omitted. This method consumes fewer resource but reduce the performance of sub-pixel precision [73]. W. CAI et al. [33] have designed a parallel and pipeline architecture for SURF algorithm. The SURF image feature point detecting system is implemented with hardware and software co-design. There are four modules in the FPGA architecture, which are

Integral Image Module, Integral Image Buffer Module, Hessian Calculation Module and Non-Maximal Suppression Module. However, we proposed an FPGA architecture including SURF detector, BRIEF descriptors and BRIEF matching. Table IV lists the comparison results between the proposed method and the method that have been published based on the FPGA.

TABLE IV

COMPARISON OF FPGA UTILIZATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | Method | FFs | LUTs | BRAMs(Kb) | DSPs |
|  | SURF+BRIEF [50] | 122000 | 88462 | 1 | 0 |
|  | OpenSURF [73] | / | 52844 | 142 | 116 |
|  | SURF [74] | 42267 | 47255 | 128 | 136 |
|  | SURF [33] | 29165 | 37592 | 68 | 178 |
|  | SURF [32] | 108581 | 179559 | 80 | 244 |
|  | SURF [78] | 35804 | 37662 | 105 | 80 |
|  | SIFT+BRIEF [27] | 30002 | 21119 | 4672 | 80 |
|  | SURF [53] | / | 107873 | 1185 | 295 |
|  | SURF+BRIEF [72] | 29541 | 25463 | 116.5 | 160 |
|  | Proposed | 298864 | 267095 | 11 | 144 |
|  |  |  |  |  |  |

*2) Speed Comparison*

Speed is one of the most import factors of on-board detection and matching. In the proposed method, the run time of the integral image, SURF detection, BRIEF descriptor, and BRIEF matching is 2.62us, 2.6ms, 2.48us, and 15.56us, respectively. The total run time is about 2.62ms, which guarantees a frame rate of 380fps with a resolution of 512×512 pixels at 100MHz. An efficient image matching system based on an FPGA chip was proposed in [27] which can be implemented with the SIFT feature detector, BRIEF descriptor and BRIEF matching for two 1280×720 images within 33ms, at 30fps. The hardware and software co-design system was proposed in [76], only the Fast-Hessian detector part of SURF has been chosen for hardware-only implementation. Generation of the SURF descriptor is handled entirely by software. The FPGA-SURF implementation achieves about 10fps at HD (1024×768 pixels) resolution, which is a necessity for real-time operation. A model which combines the modified SURF detector and BRIEF descriptor were presented and implemented in FPGA [50], which supports a throughput of 304fps for 512×512 pixels under a 100MHz. An FPGA design that combines SURF detector and Fast Retina key-point (FREAK) descriptor for real-time object detection [49] can process video frames with 800×600 pixels resolution at 60 fps. In [78], the hardware-accelerated version using the FPGA obtained execution times was 0.047 seconds for the image sequences with resolution of 640×480. In [74], an optimized FPGA-based SURF extractor was proposed, which achieved 131.36 fps for a video stream of VGA resolution at 40.355MHz. The data stream throughput of [50], [43], [27] and [78] is 80Mbps, 10Mbps, 28Mbps, and 6Mbps. The proposed system reaches a data stream throughput of 100Mbps. The comparisons of the proposed method with other investigators are shown in Tables V and VI.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 15 |

TABLE VI

COMPARISON THE PERFORMANCE OF TIME.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Method | Clock | | Resolut- | | SW | Descr- | Match-i | Speed | |  |
|  |  | (MHz) | | ion |  | /HW | iptor | ng | time | |  |
|  |  |  |  |  |
|  |  | SURF+ |  |  |  |  | HW(F |  |  |  |  |  |
|  |  | BRIEF |  | 100 | 512×512 | | Yes | Yes | 3.29ms | |  |
|  |  |  | PGA) |  |
| [50] | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SIFT |  | 100 | 640×480 | | HW(F | No | No | 31ms | |  |
| [43] | | |  | PGA) |  |
|  |  |  |  |  |  |  |  |  |
|  |  | SIFT |  |  |  |  | HW(F |  |  |  |  |  |
|  |  | +BRIEF |  | 359 | 1280×720 | | Yes | Yes | 33ms | |  |
|  |  |  | PGA) |  |
| [27] | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SURF |  | 66.7 | 640×480 | | HW | Yes | No | 47ms | |  |
| [78] | | |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | Proposed |  | 100 | 512×512 | | HW | Yes | Yes | 2.62ms | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | TABLE V | |  |  |  |  |
|  |  | COMPARISON OF THE PERFORMANCE OF FPS (SW: SOFTWARE; HW: HARDWARE) | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Method |  |  | Clock | Resolution | | fps | SW/HW | Descriptor | |  |
|  |  |  |  | (MHz) |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | SURF [33] | |  | 100 | 640×480 | | 270 | SW+HW | Yes | |  |
|  |  | SIFT+BRIEF | | | 100 | 1280×720 | | 30 | HW | Yes | |  |
| [27] | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SURF [76] | |  | / | 1024×768 | | 10 | SW+HW | Yes | |  |
|  |  | SURF+BRIEF | | | 100 | 512×512 | | 304 | HW | Yes | |  |
| [50] | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SURF+ |  |  | 100 | 800×600 | | 60 | HW | Yes | |  |
|  |  | FREAK [49] | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SURF [54] | |  | 66.7 | 640×480 | | 50 | HW | Yes | |  |
|  |  | SURF+BRIEF | | | 100 | 640×480 | | 162 | HW | Yes | |  |
| [72] | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | SURF [74] | |  | 40.355 | 640×480 | | 131.36 | HW | Yes | |  |
|  |  | SURF [79] | |  | 200 | 640×480 | | 56 | HW | Yes | |  |
|  |  | Proposed | |  | 100 | 512×512 | | 380 | HW | Yes | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

VI. CONCLUSION

An optimized FPGA-based architecture for SURF and BRIEF algorithm is proposed to select the robust ground control points for georeferencing with RS image. The pipeline and parallel structure on-chip system including a modified SURF detector and BRIEF descriptor. In the SURF detector module, WLR method is used to reduce the word length of the integral image without a loss of accuracy. MEPA method is used to compute output of the FIFO in parallel. SAS method is adopted to simplify the response of the Hessian determinant, and sliding widow is used to compute the Hessian determinant in parallel. In the BRIEF descriptor module, only a 256-bit vector memory footprint is used to store a feature point descriptor. Enhanced adder trees are employed to reduce the complexity of the BRIEF matching step.

Four pairs of remotely sensed images with different textures are applied to evaluate the performance of FPGA-based implementation. The results of experiment indicated that the proposed architecture can achieve a real-time performance with 380fps at 100MHz. The proposed algorithm combines the accuracy of SURF detectors and the rapidity of the BRIEF descriptor to obtain a quick and accurate way of matching. Hence, the combined SURF-BRIEF system has the advantages of real-time, low-power and high portability.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and the associate editor for their insightful comments and suggestions.

REFERENCES

1. D. Liu, G. Zhou, J. Huang, R. Zhang, L. Shu, X. Zhou, and C. S Xin. “On-board georeferencing using FPGA-based optimized second-order polynomial equation,” *Remote Sensing*, vol. 11, no. 2, pp.124, Jan. 2019, DOI: 10.3390/rs11020124.
2. G. Zhou, R. Zhang, N. Liu, J. Huang, and X. Zhou, “On-board ortho-rectification for images based on an FPGA,” *Remote Sensing*, vol. 9, no.9, pp. 874, Aug. 2017, DOI: 10.3390/rs9090874.
3. C. C. Leng, H. Zhang, B. Li, G. R. CAI, Z. PEI, and L. HE, “Local feature descriptor for image matching: a survey,” *IEEE Access*, vol, 7, pp. 6424-6434, Dec. 2018, DOI:10.1109/ACCESS.2018. 288-

8856.

* 1. X. Deng, Y. Huang, S. Feng, and C., Wand, “Ground control point extraction algorithm for remote sensing image based on adaptive curvature threshold,” *In 2011 IEEE International Geoscience* *and Remote Sensing Symposium*, Shanghai, China, 2008, pp.137-140. DOI: 10.1109/ETTandGRS.2008.225.
  2. H. Zhang., and Q. Hu, “Fast image matching based-on improved SURF algorithm,” *2011 International Conference on Electronics,* *Communications and Control (ICECC)*, Ningbo, China, Sep.2011, pp. 1460-1463, DOI: 10.1109/ICECC.2011.6066546
  3. F. Attneave, “Some informational aspects of visual perception,” *Psychological Review*, vol. 61, no. 3, pp. 183-193, 1954, DOI:10.1037/h0054663.
  4. T. Tuytelaars, and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Foundations and trends® in computer* *graphics and vision*, vol. 3, no. 3, pp. 177-280, Jun. 2008, DOI:10.1561/0600000017.
  5. H. P. Moravec, “Rover visual obstacle avoidance,” *Proceedings of* *the 7th International Joint Conference on Artificial Intelligence*

(*IJCAI '81*), Vancouver, BC, Canada, Aug. 1981, pp. 785-790.

DOI:10.1007/s00427-011-0383-3.

1. C. Harris, and M. Stephens, “A combined corner and edge detector,”

*Proceedings of the Alvey Vision Conference*, vol. 15, no. 50, pp.147-151, 1988, DOI:10.5244/C.2.23.

1. M. Y. I. Idris, N. B. A. Warif, N. M. Arof, A. W. A. Wahab, and Z. Razak, “Acceleration FPGA-SURF feature detection module by memory access reduction,” *Malaysian Journal of Computer Science*, vol. 32, no. 1, pp. 47-61, Jan. 2019, DOI:10.22452/mjcs.vol32no1.4.
2. S. M. Smith and J. M. Brady, “SUSAN—A new approach to low level image processing,” International journal of computer vision, vol. 23, no.1, pp.45-78, May 1997, DOI:10.1023/A:1007963824710.
3. T. Lindeberg, “Scale-space theory: A basic tool for analyzing structures at different scales,” *Journal of applied statistics*, vol. 21, no.1-2, pp. 225-270, Jun 1994, DOI.org/10.1080/757582976.
4. T. Lindeberg, “Scale-space: A framework for handling image structures at multiple scales,” *In: Proc. CERN School of Computing,* *Egmond aan Zee, The Netherlands,* Sep. 1996, pp.1-12.
5. T. Lindeberg, “Feature detection with automatic scale selection,”

*International journal of computer vision*, vol. 30, no. 2, pp. 79-116,Nov. Jul. 1998, DOI:10.1023/A:1008045108935.

1. K. Mikolajczyk, and C. Schmid, “Scale & affine invariant interest point detectors,” International journal of computer vision, vol. 60, no. 1, pp. 63-86. Oct. 2004, DOI: 10.1023/b:visi.0000027790. 02288.f2.
2. K. Mikolajczyk, and C. Schmid, “A performance evaluation of local descriptors”. IEEE Transactions on Pattern Analysis and Machine

Intelligence, Institute of Electrical and Electronics Engineers. Vol. 27, no.10, pp. 1615-1630, Feb. 2005, DOI: 10.1109/TPAMI. 2005.188

1. D. G. Lowe, Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, vol. 60, no. 2, pp. 91-110,Jan. 2004, DOI:10.1023/B:VISI.0000029664.99615.94.
2. C. Evans, “Notes on the Opensurf library,” University of Bristol, Tech. Rep. CSTR-09-001, Jan. 2009.
3. Y. Ke, and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *Proceedings of the 2004*

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 16 |

*IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society*, Washington, DC,USA, vol. 2, pp. 506–513, Jul. 2004, DOI:10.1109/CVPR. 2004.1315206.

1. E. N. Mortensen, H. Deng, and L. Shapiro, “A SIFT descriptor with global context,” *2005 IEEE Computer Society Conference on* *Computer Vision and Pattern Recognition*, San Diego, CA, USA,2005, pp.184-190, DOI: 10.1109/CVPR.2005.45.
2. A. E. Abdel-Hakim, and A. A. Farag. “CSIFT: A SIFT descriptor with color invariant characteristics,” *In Computer Vision and Pattern* *Recognition* (*CVPR 2006*), New York, USA, 2006, pp.1978-1983.DOI: 10.1109/CVPR.2006.95.
3. J. M. Morel, and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol. 2 no. 2, pp. 438-469, Apr. 2009, DOI: 10.1137/080732730.
4. H. Bay, T. Tuytelaars, and L.V. Goo, “SURF: Speeded up robust features,” *In European conference on computer vision*. Berlin, Heidelberg, 2006, Part II, pp. 404-417, DOI: 10.1007/11744023\_32.
5. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer vision and image understanding*. Vol. 110, no. 3, pp. 346-359, Jun. 2008. DOI: 10.1016/j.cviu.2007. 09.014
6. S. Leutenegger, M. Chli, and R. Y. Siegwart, “RISK: Binary robust invariant scalable keypoints,” *In 2011 IEEE international* *conference on computer vision*, Barcelona, Spain, Nov. 2011, pp.2548-2555, DOI: 10.1109/ICCV.2011.6126542.
7. M. Calonder, V. Lepetit, M. Oezuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a Local Binary Descriptor Very Fast,”

*IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol. 34, no. 7, pp. 1281–1298, Jul. 2012, DOI: 10.1109/TPAMI. 2011.222.

1. J. H. Wang, S. Zhong, W. H. Xu, W. J. Zhang, and Z. G. Cao, “A FPGA-based architecture for real-time image matching,” *MIPPR* *2013: Parallel Processing of Images and Optimization and Medical Imaging Processing*, *International Society for Optics and Photonics*,vol. 8920, pp. 892003, Oct. 2013, DOI: 10.1117/12.2031050.
2. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” *IEEE international* *conference on computer vision*, Barcelona, Spain, Nov. 2011, pp.2564-2571, DOI: 10.1109/ICCV.2011.6126544.
3. E. Rosten, and T. Drummond, “Fusing points and lines for high performance tracking,” *Tenth IEEE international conference on* *computer vision*, 17-21 Beijing, China, Oct. 2005, pp. 1508-1515,DOI:10.1109/ICCV.2005.104.
4. R. D. Lima, J. Martinez-Carranza, A. Morales-Reyes, and R. Cumplido. “Accelerating the construction of BRIRF descriptors using an FPGA-based architecture,” *2015 International Conference* *on ReConFigurable Computing and FPGAs*, Mexico City, Mexico,2015, pp. 1-6, DOI: 10.1109/ReConFig.2015.7393285.
5. A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” *2012 IEEE Conference on Computer Vision and Pattern* *Recognition*, Providence, RI, USA, 2012. pp. 510-517. DOI:10.1109/CVPR.2012.6247715.
6. J. Zhao, S. Zhu, and X. Huang, “Real-time traffic sign detection using SURF features on FPGA,” *In 2013 IEEE high performance extreme* *computing conference* (*HPEC*), *IEEE*, Waltham, MA, USA, 2013,
   1. 1-6, DOI: 10.1109/HPEC.2013.6670350.
7. W. CAI, Z. XU, and Z. LI, “A high performance SURF image feature detecting system based on ZYNQ,” *DEStech Transactions on* *Computer Science and Engineering*, vol. 12, pp.256-261, 2017,dio:10.12783/dtcse/cii2017/17261.
8. G. Zhou and R. Li. “Accuracy evaluation of ground points from IKONOS high-resolution satellite imagery,” *Photogrammetry* *Engineering & Remote Sensing*, vol. 66, no. 9, pp. 1103-1112, Sep.2000, DOI:10.1016/S0924-2716(00)00020-4.
9. S. S. Cai, L. B. Liu, S. Y. Yin, R. Y. Zhou, W. L. Zhang, and S. J. Wei, “Optimization of speeded-up robust feature algorithm for hardware implementation,” *Science China Information Sciences*, vol. 57, no. 4,
   1. 1-15, Jan. 2014, DOI: 10.1007/s11432-013-4946-y.
10. G. Hua, M. Brown, and S. Winder, “Discriminant embedding for local image descriptors,” *In 2007 IEEE 11th International Conference on* *Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1-8,10.1109/ICCV.2007.4408857.
11. R. Rani, A. P. Singh, and R, Kumar. “Impact of reduction in descriptor size on object detection and classification”, *Multimedia Tools and*

*Applications*, vol. 78, no.7, pp. 8965-8979, Nov. 2018, DOI:

10.1007/s11042-018-6911-7.

1. M. Calonder, V. Lepetit, P. Fua, K. Konolige, J. Bowman, and P. Mihelich, “Compact signatures for high-speed interest point description and matching,” *In 2009 IEEE 12th International* *Conference on Computer Vision*, Kyoto, Japan, 2009, pp. 357-364.DOI: 10.1109/ICCV.2009.5459272.
2. H. Je´gou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” IEEE Trans. *Pattern Analysis and* *Machine Intelligence*, vol. 33, no.1, pp. 117-128, Jan. 2011, DOI:10.1109/TPAMI.2010.57.
3. J. Huang, G. Zhou, X. Zhou, and R. Zhang. “A new FPGA architecture of FAST and BRIEF algorithm for on-board corner detection and matching,” Sensors, vol. 18, no. 4, pp.1014, Mar. 2018. DOI: 10.3390/s18041014.
4. P. Čížek, F. Jan, and D. Masri. “Low-latency image processing for vision-based navigation systems,” *In 2016 IEEE International* *Conference on Robotics and Automation (ICRA),* Stockholm,Sweden, 2016, pp. 781-786. DOI: 10.1109/ICRA.2016.7487207.
5. T. Krajník, J. Šváb, S. Pedre, P. Čížek, and L. Přeučil, “FPGA-based module for SURF extraction,” *Machine vision and applications*, vol. 25, no. 3, pp. 787-800, Feb. 2014, DOI: 10.1007/s00138-014 -0599-0.
6. L. Yao, H. Feng, Y. Zhu, Z. Jiang, D. Zhao, and W. Feng, “An architecture of optimised SIFT feature detection for an FPGA implementation of an image matcher,” *2009 International* *Conference on Field-Programmable Technology. Sydney*, NSW,Australia, 2009, pp. 30-37. DOI:10.1109/FPT.2009.5377651.
7. D. Kim, M. Kim, K. Kim, M. Sung, and W. W. Ro. “Dynamic load balancing of parallel SURF with vertical partitioning,” *IEEE* *Transactions on Parallel and Distributed Systems*. vol. 26, no. 12,pp. 3358-3370, Dec. 2014, DOI: 10.1109/TPDS.2014.2372763.
8. S. H. Cheon, I. K. Eom, and Y. H. Moon, “Fast descriptor extraction method for a SURF-based interest point,” *Electronics Letters*, vol. 52, no. 4, pp. 274-275, Feb. 2016, dio:10.1049/el.2015.3055.
9. J. Kim, E. Park, X. Cui, and H. Kim, “A fast feature extraction in object recognition using parallel processing on CPU and GPU.”

*2009 IEEE International Conference on Systems*, San Antonio, TX,USA, 2009, pp. 3842-3847. DOI: 10.1109/ICSMC.2009.5346612.

1. M. Schaeferling, U. Hornung, and G. Kiefer, “Object recognition and pose estimation on embedded hardware: SURF-based system designs accelerated by FPGA logic,” *International Journal of* *Reconfigurable Computing*, vol. 2012, no.6, pp. 1-16, Jan. 2012,DOI: 10.1155/2012/368351.
2. J. Huang, G. Zhou, D. Zhang, G. Zhang, R. Zhang, and O. Baysal, “An FPGA-based implementation of corner detection and matching with outlier rejection,” *International journal of remote sensing*, vol. 39,

no. 23, pp. 8905-8933, Aug. 2018, DOI: 10.1080/01431161.2018.1500728.

1. J. Zhao, X. Huang, and Y. Massoud, “An efficient real-time FPGA implementation for object detection,” *IEEE 12th International New* *Circuits and Systems Conference* (*NEWCAS*). Trois-Rivieres, QC,Canada, 2014, pp. 313-316. DOI: 10.1109/NEWCAS.2014. 6934045.
2. J. Huang, and G. Zhou, “On-board detection and matching of feature points,” *Remote Sensing*, vol. 9, no. 6, pp. 601. Apr. 2017, DOI: 10.3390/rs9060601.
3. A. Li, W. Jiang, W. Yuan, D. Dai, S. Zhang, and Z. Wei, “An improved FAST+ SURF fast matching algorithm,” *Procedia Computer* *Science*, vol. 107, pp. 306-312, Dec. 2017, DOI: 10.1016/j.procs.2017.03.110.
4. T. Kasezawa, H. Tanaka, and H. Ito. “Integral image word length reduction using overlapping rectangular regions” *2016 IEEE* *International Conference on Industrial Technology* (*ICIT*). Taipei,Taiwan, 2016, pp. 763-768. DOI: 10.1109/ICIT.2016.7474847.
5. X. Fan, C. Wu, W. Cao, X. Zhou, S. Wang, and L. Wang. “Implementation of high performance hardware architecture of OpenSURF algorithm on FPGA,” *in 2013 International Conference* *on Field-Programmable Technology* (*FPT*), Kyoto Japan, 2013, pp.152-159, DOI: 10.1109/FPT.2013.6718346.
6. M. Pohl, M. Schaeferling and G. Kiefer, “An efficient FPGA-based hardware framework for natural feature extraction and related Computer Vision tasks,” in *2014 24th International Conference on* *Field Programmable Logic and Applications* (FPL), Munich,Germany, 2014, pp. 1-8. DOI: 10.1109/FPL.2014.6927463.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 17 |

1. M. Schaeferling, and G. Kiefer, “Flex-SURF: a flexible architecture for FPGA-based robust feature extraction for optical tracking systems,”in: *IEEE International Conference on Reconfigurable* *Computing and FPGAs* (*ReConFig*), Quintana Roo, Mexico, 2010.pp. 458-463. DOI: 10.1109/ReConFig.2010.11.
2. T. B. Terriberry, L. M. French, and J. Helmsen, “GPU accelerating speeded-up robust features,” in *Proceedings of 3DPVT*, vol. 8, pp. 355-362, Jun. 2008, DOI: 10.1016/j.cviu.2007.09.014.
3. E. Oyallon, and J. Rabin, “An analysis of the SURF method,” *Image* *Processing Online*, vol. 5, pp. 176-218, Jul. 2015, dio:10.5201/ipol.2015.69.
4. N. Zhang, “Computing optimised parallel speeded-up robust features (p-surf) on multi-core processors,” *International journal of parallel* *programming*, vol. 38, no. 2, pp. 138-158, Apr. 2010, DOI10.1007/s10766-009-0122-9.
5. Gossow, David, Peter Decker, and Dietrich Paulus, “An evaluation of open source SURF implementations,” *Robot Soccer World Cup*, Springer, Berlin, Heidelberg, 2010, pp.169-179. DOI: 10.1007/978-3-642-20217-9\_15.
6. Chris Evans, Available online. https://ww2.mathworks.cn/ matlabcentral/fileexchange/28300-opensurf-including-image-warp.
7. P. H. Hsu, and S. Y. Chien, “Reconfigurable cache memory architecture for integral image and integral histogram applications,” in *2011 IEEE Workshop on Signal Processing Systems* (*SiPS*). Beirut, Lebanon, 2011, pp. 151-156. DOI: 10.1109/SiPS.2011.6088966.
8. H. J. W. Belt, “Word length reduction for the integral image,” i*n* *Proceedings of the 15th IEEE International Conference on Image Processing*, San Diego, CA, USA, 2008, pp. 805–808. DOI:10.1109/ICIP.2008.4711877.
9. S. H. Lee, and Y. J. Jeong, “A new integral image structure for memory size reduction,” *IEICE TRANSACTIONS on Information* *and Systems*, vol. 97, no. 4, pp. 98-1000, Apr. 2014, DOI:10.1587/transinf.E97.D.998.
10. S. Ehsan, and K. D. McDonald-Maier, “Exploring integral image word length reduction techniques for SURF detector,” in *2009* *Second International Conference on Computer and Electrical Engineering, IEEE*. Dubai, United Arab Emirates, Dubai, UnitedArab Emirates, 2009, pp. 635-639, DOI: 10.1109/ICCEE.2009.138.
11. S. Ehsan, A. F. Clark, and K. D. McDonald-Maier, “Novel hardware algorithms for row-parallel integral image calculation,” in *2009* *Digital Image Computing: Techniques and Applications*, *IEEE*,Melbourne, VIC, Australia, 2009. pp. 61-65, DOI: 10.1109/DICTA. 2009.20.
12. S. Ehsan, A. F. Clark, N. Rehman, and K. McDonald-Maier, “Integral images: efficient algorithms for their computation and storage in resource-constrained embedded vision systems,” *Sensors*, 2015, 15(7), 16804-16830. DOI:10.3390/s150716804.
13. P. Viola, and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *proceedings of the IEEE computer* *society conference on computer vision and pattern recognition*,Kauai, HI, USA, 2001, pp.511-518. DOI: 10.1109/CVPR.2001. 990517.
14. B. Kisacanin, “Integral Image Optimizations for Embedded Vision Applications,” in *Proceedings of the IEEE Southwest Symposium on* *Image Analysis and Interpretation*, Santa Fe, NM, USA, 2008, pp.181–184. DOI: 10.1109/SSIAI.2008.4512315.
15. T. Sledeviˇc, and A. Serackis, “SURF algorithm implementation on FPGA,” in *Proceedings of the 2012 13th Biennial Baltic Electronics* *Conference*, Tallinn, Estonia; 2012, pp. 291–294. DOI: 10.1109/BEC. 2012.6376874.
16. M. Schaeferling, and G. Kiefer, “Object Recognition on a Chip: A Complete SURF-based System on a Single FPGA,” *International* *Conference on Reconfigurable Computing and FPGAs*, Cancun,Mexico, 2011, pp.49-54. DOI: 10.1109/ReConFig.2011.65.
17. M. Schmidt, M. Reichenbach, and D. Fey, “Generic VHDL Template for 2D Stencil Code Applications on FPGAs,” in *2012 IEEE 15th* *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, Shenzhen,Guangdong, China, 2012; pp. 180–187. DOI: 10.1109/ISORCW. 2012.39.
18. Q. Ni, F. Wang, Z. Zhao, and P. Gao, “FPGA-based Binocular Image Feature Extraction and Matching System,” *arXiv preprint* *arXiv*:1905.04890 [cs. CV], May 2019, DOI:10.1117/12.2205633.
19. C. Chen, H. Yong, S. Zhong, and L. Yan, “A real-time FPGA-based architecture for OpenSURF,” *Proc. SPIE 9813, MIPPR 2015:*

*Pattern Recognition and Computer Vision*, vol. 9813, pp. 98130k,Dec. 2015, DOI: 10.1117/12.2205633.

1. C. Wilson, P. Zicari, S. Craciun, P. Gauvin, E. Carlisle, A. George, and H. Lam, “A power-efficient real-time architecture for SURF feature extraction,” in *2014 International Conference on ReConFigurable* *Computing and FPGAs*, 2014, Cancun, Mexico, pp. 1-8.DOI:10.1109/ReConFig.2014.7032492.
2. P. Čížek, and J. Faigl, “Real-Time FPGA-Based Detection of Speeded-Up Robust Features Using Separable Convolution”, *IEEE* *Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1155-1163.Mar. 2017, DOI: 10.1109/TII.2017.2764485.
3. J. Svab, T. Krajnik, J. Faigl, and L. Preucil, “FPGA based speeded up robust features,” *in 2009 IEEE International Conference on* *Technologies for Practical Robot Applications*, 2009, Woburn, MA,USA, pp. 35–41, DOI: 10.1109/TEPRA.2009.5339646.
4. M. Fularz, M. Kraft, A. Schmidt, and A. Kasiński, “A high-performance FPGA-based image feature detector and matcher based on the FAST and BRIEF algorithms”, *International Journal of* *Advanced Robotic Systems*, vol. 12, no. 10, pp. 141, Oct. 2015, DOI:10.5772/61434.
5. W. Chen, S. Ding, Z. Chai, D. He, W. Zhang, G. Zhang, and W. Luo, “FPGA-based parallel implementation of SURF algorithm,” *in 2016* *IEEE 22nd International Conference on Parallel and Distributed Systems* (*ICPADS*), Wuhan, China, 2016. pp. 308-315. DOI:10.1109/ICPADS.2016.0049.
6. D. Bouris, A. Nikitakis, and I. Papaefstathiou, “Fast and efficient FPGA-based feature detection employing the SURF algorithm,” in *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, *IEEE*,Charlotte, NC, USA, 2010, pp. 3-10. DOI: 10.1109/FCCM.2010.11.
7. S. Ehsan, N. Kanwal, A. F. Clark, and K. D. McDonald-Maier, “An algorithm for the contextual adaption of surf octave selection with good matching performance: Best octaves,” *IEEE Transactions on* *Image Processing*, vol. 21, no. 1, pp. 297-304, Jan. 2012, DOI:10.1109/TIP.2011.2160869.
8. J. Li, T. Xu, and K. Zhang, “Real-time feature-based video stabilization on FPGA.” *IEEE Transactions on Circuits and Systems* *for Video Technology*, vol. 27, no.4, pp. 907-919, Jan. 2016, DOI:10.1109/TCSVT.2016.2515238.

**Dequan LIU** received the B.S. degree inElectronic Science and Technology from Northwest Normal University, Lanzhou, China, in 2004 and the M.E. degree in Signal and Information Processing from Xidian University, Xi’an, China, in 2013. He is currently pursuing the Ph.D. degree in Circuits and Systems at School of Microelectronics, Tianjin University,



Tianjin, China.

**Guoqing Zhou** (M’02–SM’05) receivedthe Ph.D. degree from Wuhan University, Wuhan, China, in 1994. He was a Visiting Scholar with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, and a Postdoctoral Researcher with the Institute of Information Science, Beijing Jiaotong



University, Beijing. He continued his research as an Alexander von Humboldt Fellow with the Technical University of Berlin, Berlin, Germany, and afterward became a Postdoctoral Researcher with The Ohio State University, Columbus, OH, USA, from 1998 to 2000. He later had been with Old Dominion University, Norfolk, VA, USA, as an Assistant Professor, Associate Professor, and Full Professor in 2000, 2005, and 2010, respectively. He is currently with Guilin University of Technology, Guilin, China, as a “1000 Plan Talent Scholarship

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTARS.2020.2998838, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

|  |  |
| --- | --- |
| IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING | 18 |

Professor” honored by the Chinese Government. He has

published 2 books and more than 220 referred publications. He

has worked on 48 projects as a Principal Investigator or a

Co-Principal Investigator and participant as a Postdoctoral

Researcher.



**Dianjun Zhang** received the B.S. and M.S.

degrees in geographical information system

from Ludong and Beijing Forestry University,

China, in 2008 and 2011, respectively and

received the Ph.D. degree at the Institute of

Geographic Sciences and Natural Resources

Research, Chinese Academy of Sciences,

Beijing, China.

His research interests include thermal

infrared remote sensing and retrieval of surface parameters

from satellite data.

**Xiang Zhou** received the M.S. degree in the



Detection Technology and Automatic

Equipment from Guilin University of

Technology in 2014. He is currently with the

School of Microelectronics, Tianjin

University, Tianjin, China. His research

interest is LiDAR imaging.



**Chenyang Li** was born in Hebei, China, in

1990. He is currently working towards his

Ph.D. degree in the School of Marine Science

and Technology, Tianjin University.

His research interests include airborne Lidar

waveform data Processing, the character of

Blue-Green laser transmission through rough

sea surface, the methods and applications of

airborne Lidar bathymetry.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.