electronics

MDPI

Article

FPGA Design of Enhanced Scale-Invariant Feature Transform with Finite-Area Parallel Feature Matching for Stereo Vision

Chien-Hung Kuo, Erh-Hsu Huang, Chiang-Heng Chien and Chen-Chien Hsu *

Department of Electrical Engineering, National Taiwan Normal University, Taipei 10610, Taiwan; chk@ntnu.edu.tw (C.-H.K.); 60675013h@ntnu.edu.tw (E.-H.H.); chiangheng.chien@gmail.com (C.-H.C.)
* Correspondence: jhsu@ntnu.edu.tw; Tel.: +886-2-7749-3551

Abstract: In this paper, we propose an FPGA-based enhanced-SIFT with feature matching for stereo vision. Gaussian blur and difference of Gaussian pyramids are realized in parallel to accelerate the processing time required for multiple convolutions. As for the feature descriptor, a simple triangular identification approach with a look-up table is proposed to efficiently determine the direction and gradient of the feature points. Thus, the dimension of the feature descriptor in this paper is reduced by half compared to conventional approaches. As far as feature detection is concerned, the condition for high-contrast detection is simplified by moderately changing a threshold value, which also benefits the reduction of the resulting hardware in realization. The proposed enhanced-SIFT not only accelerates the operational speed but also reduces the hardware cost. The experiment results show that the proposed enhanced-SIFT reaches a frame rate of 205 fps for 640 × 480 images. Integrated with two enhanced-SIFT, a finite-area parallel checking is also proposed without the aid of external memory to improve the efficiency of feature matching. The resulting frame rate by the proposed stereo vision matching can be as high as 181 fps with good matching accuracy as demonstrated in the experimental results.

Keywords: SIFT; FPGA; feature detection; feature descriptor; feature matching; stereo vision; Epipolar

Citation: Kuo, C.-H.; Huang, E.-H.; Chien, C.-H.; Hsu, C.-C. FPGA Design of Enhanced Scale-Invariant Feature Transform with Finite-Area Parallel Feature Matching for Stereo Vision. *Electronics* **2021**, *10*, 1632. https://doi.org/10.3390/electronics10141632

Academic Editors: Stefano Ricci and Akash Kumar

Received: 15 May 2021
Accepted: 7 July 2021
Published: 8 July 2021

1. Introduction

Recently, vision-based simultaneous localization and mapping (V-SLAM) techniques have become more and more popular due to the need for the autonomous navigation of mobile robots [1,2]. The front-end feature point detection and feature matching are especially important because their accuracy will significantly influence the performance of back-end visual odometry, mapping, and pose estimation [3,4]. In the front-end schemes, although speed-up robust features (SURF) exhibit a faster operational speed, its accuracy is worse than scale-invariant feature transform (SIFT) [5,6]. Nevertheless, the high accuracy of SIFT is achieved at the cost of a time-consuming process. Although CUDA implementations on GPUs for parallel programming can be used for various feature detection and matching [7,8], the software-based approaches generally require a larger power consumption which is not desired for mobile robot vision applications. Thus, this paper aims to improve the operational speed of SIFT by hardware implementation without sacrificing its accuracy. Over the past years, many researchers were devoted to improving the operational efficiency of SIFT [9]. Among them, a pipelined FPGA-based architecture was proposed in [10] to process images with double speed at the price of a higher hardware cost. A hardware implemented SIFT was also proposed in [11] to reduce the number of internal registers. However, the resulting image frame rate is not high enough because of finite bandwidth due to external memory. As a result, the efficiency of the subsequent mapping process could be limited. In [12], multiple levels of

Gaussian-blurred images were simultaneously generated by adequately modifying Gaussian kernels to shorten the processing time, resulting in a frame rate of 150 fps for 640 × 480 images. Although a few dividers were only used by the modified approach for feature detection, the resulting hardware cost is difficult to reduce because of the square function in the design. Besides, the coordinate rotation digital computer (CORDIC) algorithm adopted in finding phases and gradients of the feature descriptor often requires considerable hardware resources and a long latency period due to the needs of multiple iterations, thus significantly preventing the approach from being applied for real-world applications.

As far as feature matching is concerned, one feature point of an image needs to compare with all feature points of the other image to find the matching pairs [13] in conventional exhaustion methods. A large number of feature points would incur extra memory access time. Moreover, the realization of feature matching often consumes a large number of hardware resources. Although the random sample consensus (RANSAC) algorithm [14] can improve the accuracy of feature matching, the resulting frame rate is only 40 fps. Therefore, it has brought a great challenge to improve the operational speed of the SIFT for use with feature matching.

In this paper, we propose an enhanced-SIFT (E-SIFT) to reduce the hardware resources required and accelerate the operational speed without sacrificing accuracy. Gaussian-blurred images and difference of Gaussian (DoG) pyramids can be simultaneously obtained by meticulous design of a realizing structure. We also propose a simple triangular identification (TI) with a look-up table (LUT) to easily and quickly decide the direction and gradient of a pixel point. The resulting dimension of feature points can thus be effectively reduced by half. This is not only beneficial to the operational speed of the proposed E-SIFT and the following feature matching but also helpful to reduce the hardware cost. In the feature detection, the condition of high-contrast detection is also modified by slightly adjusting a threshold value without substantial change, thus considerably reducing the required logic elements and processing time.

For stereo vision, we also propose a finite-area parallel (FAP) feature matching approach integrated with two E-SIFTs. According to the position of the feature point of the right image, a fixed number of feature points in the corresponding area of the left image will be chosen and stored first. Then, feature matching proceeds in parallel to efficiently and accurately find the matching pairs without using any external memory. Based on the Epipolar geometry, a minimum threshold is assigned during feature comparison to avoid matching errors caused by the visual angle difference between the two cameras [15]. Besides, a valid signal is designed in the system to prevent the finite bandwidth of external SDRAM or discontinuous image data transmission from corrupting the matching accuracy. In the experiments, we use Altera FPGA hardware platform to evaluate the feasibility of the proposed TI with LUT scheme and test the flexibility of the proposed E-SIFT and double E-SIFT with FAP feature matching.

2. Proposed E-SIFT and Fap Feature Matching for Stereo Vision

Figure 1 shows the proposed FPGA-based E-SIFT architecture, where an initial image is convoluted by a meticulous design of Gaussian filters to simultaneously generate a Gaussian-blurred image and DoG pyramid. Through a definite mask in the Gaussian-blurred image, the direction and gradient of each detection point can be determined by the proposed TI with LUT method. The results in each local area are then summed and collected to form a feature descriptor. Since the summations among different partition areas present large differences, values in the feature descriptors will be normalized to acceptable ones by a simple scheme without considerably altering the original character of distribution. At the same time, on the right side in Figure 1, the corresponding detection point in the DoG images will pass through extrema, high contrast, and corner detections to decide whether feature points exist. If the result is positive, a confirmed signal and the corresponding feature descriptor will be delivered to the feature matching module.

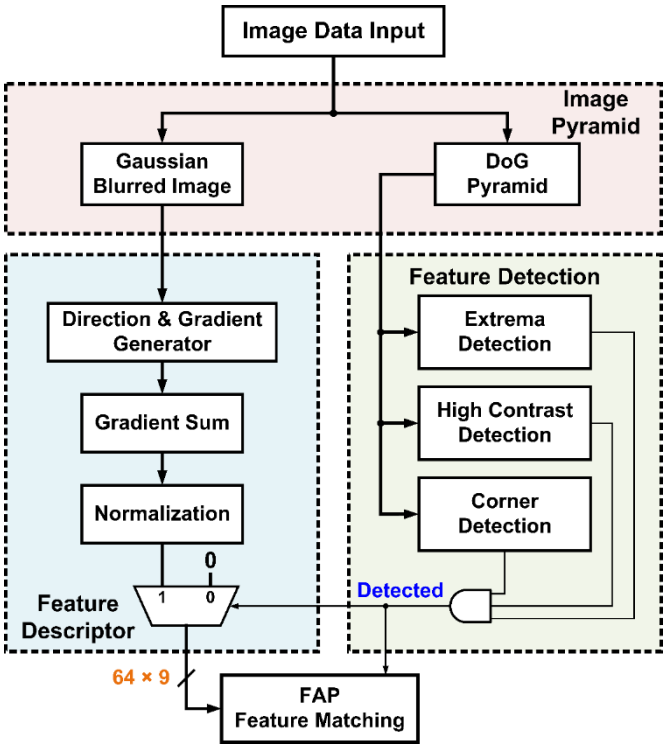


Figure 1. Building blocks of the proposed E-SIFT.

2.1. Image Pyramid

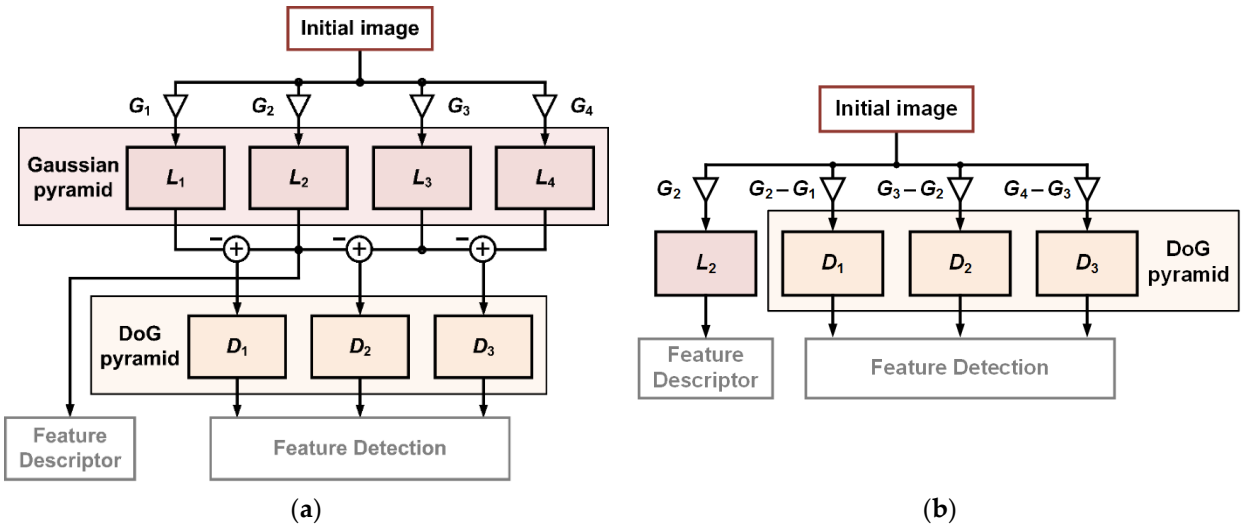
The purpose of building the image pyramid is to find out feature points from the differences of consecutive scale-spaces. Multi-level Gaussian-blurred images are created one by one through a series of Gaussian functions. The speckles and interfering noises in the original image can therefore be reduced [9]. Then, the contour of objects in the original image is vaguely outlined by the differences between the consecutive Gaussian-blurred images. Thus, the image feature can be reserved to be the basis for subsequent feature detection. In general, long latency is expected in constructing the multi-level Gaussian-blurred pyramids. Thus, as shown in Figure 2a, a more efficient structure had been proposed by simultaneously convoluting all the Gaussian functions, where suitable Gaussian kernels can be found by training the system with different images [12]. To further accelerate the construction of the DoG pyramid, a simplified structure is proposed in this paper, where new kernels for convolution are first derived by subtraction of two adjacent ones as shown in Figure 2b. Then, the nth level DoG image, $D_n(x, y)$, can be obtained by directly convoluting the original image with the new kernel:

$$D_n(x,y) = L_{n+1}(x,y) - L_n(x,y) = [G_{n+1}(x,y) - G_n(x,y)] * I(x,y), \quad (1)$$

where $L_n(x, y)$ and $G_n(x, y)$ are the nth level Gaussian-blurred image and Gaussian kernel, respectively, and $I(x, y)$ is the initial image. In the proposed structure, the DoG pyramid and the required Gaussian-blurred image can be directly created at the same time. Here, only the 2nd level Gaussian-blurred image is reserved to be the basis of the feature descriptor.

Since a 7 × 7 mask is adopted mainly for the convolution in the image pyramid, 49 general-purpose 8-bit registers will be required to store the pixel data. For an image width of 640 pixels, a large number of 8-bit registers needs be aligned row by row for the serial input of pixel data. To dramatically reduce the number of internal registers, we propose a buffer structure as shown in Figure 3. We utilize Altera optimized RAM-based shift registers to form a six-row buffer array, where each row includes 633 registers. Once the pixel data reaches the last one, i.e., reg. 49, the 7 × 7 convolution can be accomplished in one clock cycle. A similar buffer structure with different sizes for the following building blocks

requiring a mask can also be adopted to efficiently reduce the register number as well as shorten the buffer delay.



(**a**) (**b**)

Figure 2. (**a**) Conventional structure and (**b**) the proposed structure in realizing image pyramids.

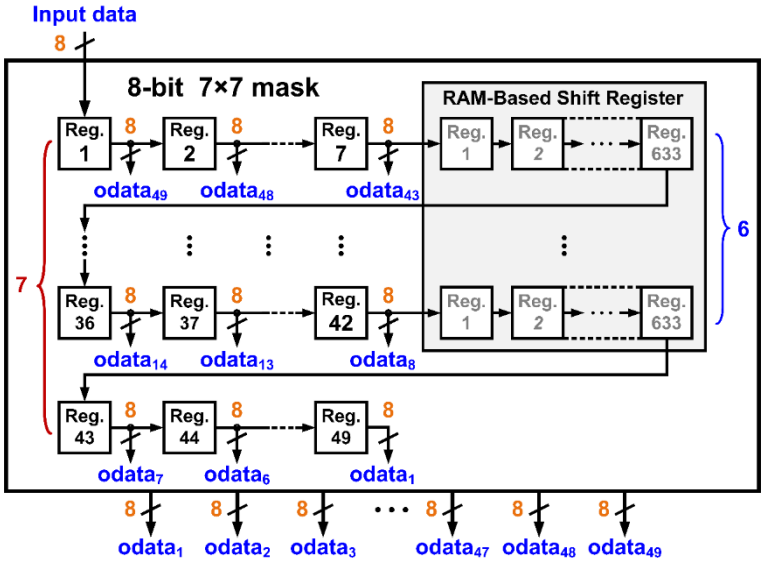


Figure 3. 8-bit input buffers for 7 × 7 mask.

2.2. Feature Descriptor

To accelerate the generating speed of feature descriptors, the direction and gradient of the detection point need to be quickly determined. Thus, in this paper, the TI method is proposed to promptly determine the direction of the detection point. Combined with a look-up table, the gradient of the detection point can be easily estimated. In general, eight different directions need to be determined to reduce the complexity of the phase of the detection point [9,16]. In this paper, the eight directions with unsigned gradients for the detection point have been simplified to four with signed ones to reduce the hardware cost. Although the resulting gradient size increases by one bit, the dimension of the feature descriptor can be effectively reduced by half. It is also helpful to the FAP feature matching for implementation.

Figure 4 shows the block diagram of the feature descriptor. First, a 3 × 3 mask is applied to the 2nd level Gaussian-blurred image to find the differences between adjacent pixels in x and y axes to the center. The direction and gradient of the center point will then

be determined respectively by the proposed TI with LUT method. Finally, the feature descriptor of the detection point can be synthesized by finding and collecting gradient sums of the four directions in each 4 × 4 partition area of a 16 × 16 mask.

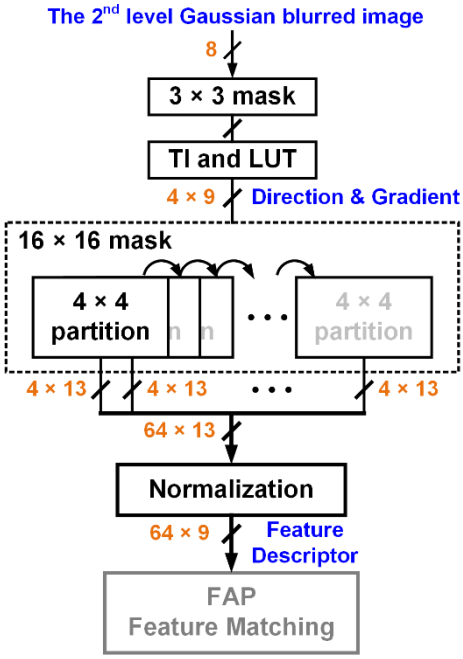


Figure 4. Function blocks of the feature descriptor.

2.2.1. Triangular Identification and Look-Up Table

In conventional approaches, the center point, i.e., $L_2(x, y)$, of every 3 × 3 mask shifting in the 2nd level Gaussian-blurred image is the detection point. The variations of the detection point on x and y axes, i.e., Δp and Δq, are defined by the differences between two adjacent pixel values, as shown in Figure 5.

$$\Delta p = L_2(x+1, y) - L_2(x-1, y) \tag{2}$$

$$\Delta q = L_2(x, y+1) - L_2(x, y-1) \tag{3}$$

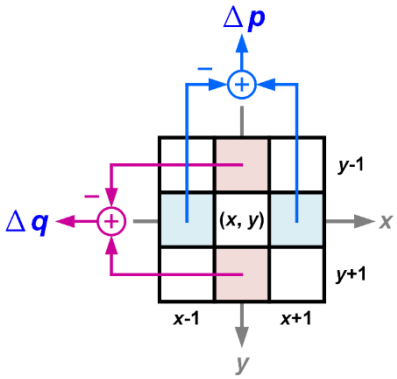


Figure 5. Variations of the detection point on x and y axes in a 3 × 3 mask.

Since these two variations are perpendicular to each other, the resulting phase and magnitude which define the direction and gradient of the detection point can be easily determined by the following equations [9].

$$\theta(x,y) = \arctan\left(\frac{\Delta q}{\Delta p}\right) \tag{4}$$

$$m(x,y) = \sqrt{\Delta p^2 + \Delta q^2} \tag{5}$$

Conventional approaches generally utilize the CORDIC algorithm to derive the results of Equations (4) and (5) [17]. However, time-consuming and tedious procedures would ensue due to a large number of iterations. In this paper, we propose a simple TI method to determine the phase of detection points. Since only eight directions exist in the original definition, the direction of the detection point can be easily determined by the value of Δp and Δq, which are two legs of a right-angled triangle. For example, if Δp is larger than Δq, and both are greater than zero, then the hypotenuse will be located in direction 0, as shown in Figure 6a. Thus, the direction of the detection point is classified as phase 0 directly. Conversely, the direction of the detection point is classified as phase 1 if Δp is smaller than Δq. In a similar way, we can easily recognize other different directions by using Table 1. The required conditions are also devised to distinguish different directions. Absolute values are considered in reality because the lengths of both legs cannot be negative.

For a right-angled triangle, if there is a big difference between the length of the two legs, the hypotenuse length can be approximated as the longer one. If two legs have equal lengths, i.e., $\Delta p = \Delta q$, the hypotenuse length will be $\sqrt{2}$ times leg one, as shown in Figure 6b. Thus, the gradient of the detection point cannot be directly estimated only by the comparison of the two legs. Here, a look-up table is proposed for the estimation of the gradient of the detection point, as listed in Table 2. First, the ratio, h, of variations in two axes is evaluated for choosing the correction factor, K, as shown in the table. The gradient of the detection point, m, will then be approximated by the length of the longer leg multiplied by a chosen correction factor.

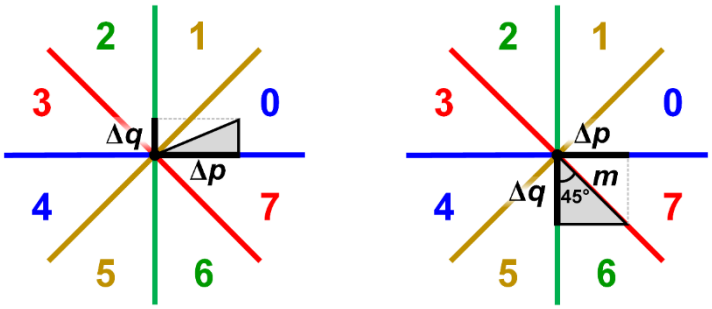


Figure 6. (**a**) The proposed TI algorithm for direction recognition; (**b**) The ratio of hypotenuse to the leg has maximum value when $\Delta p = \Delta q$.

Table 1. Angle range condition and direction distribution.

Direction	Conditions	Pixel Phase θ
0	$\Delta p > 0, \Delta q \geq 0$ & $\lvert\Delta p\rvert > \lvert\Delta q\rvert$	$0° \leq \angle\theta < 45°$
1	$\Delta p > 0, \Delta q > 0$ & $\lvert\Delta p\rvert \leq \lvert\Delta q\rvert$	$45° \leq \angle\theta < 90°$
2	$\Delta p \leq 0, \Delta q > 0$ & $\lvert\Delta p\rvert < \lvert\Delta q\rvert$	$90° \leq \angle\theta < 135°$
3	$\Delta p < 0, \Delta q > 0$ & $\lvert\Delta p\rvert \geq \lvert\Delta q\rvert$	$135° \leq \angle\theta < 180°$
4	$\Delta p < 0, \Delta q \geq 0$ & $\lvert\Delta p\rvert > \lvert\Delta q\rvert$	$180° \leq \angle\theta < 225°$
5	$\Delta p < 0, \Delta q < 0$ & $\lvert\Delta p\rvert \leq \lvert\Delta q\rvert$	$225° \leq \angle\theta < 270°$
6	$\Delta p \geq 0, \Delta q < 0$ & $\lvert\Delta p\rvert < \lvert\Delta q\rvert$	$270° \leq \angle\theta < 315°$
7	$\Delta p > 0, \Delta q < 0$ & $\lvert\Delta p\rvert \geq \lvert\Delta q\rvert$	$315° \leq \angle\theta < 360°$

Note that the feature descriptor requires eight dimensions to express different gradients of the eight directions. To reduce hardware resources, a representation having fewer dimensions is discussed in this paper. From the observation of Figure 6, phase 4 has an opposite direction to phase 0. Similarly, phase 5 has an opposite direction to phase 1, and so on. Thus, the eight-direction representation can be simplified to a four-direction one to reduce the resulting dimensions. For example, if the detection point has an unsigned gradient in phase 4, it will be changed to have a negative one in phase 0. As shown in Figure 7, the proposed TI with LUT approach for detecting the direction and gradient of the detection point is processed in parallel to accelerate the operational speed. The presented structure is simpler than the CORDIC, so that the hardware resources for implementation can be effectively reduced.

Table 2. Look-up table for gradient correction.

$h = \lvert\Delta p/\Delta q\rvert$	Correction Factor (K)	Pixel Gradient (m)
$h \le 0.25$	1.00	
$0.25 < h \le 0.52$	1.08	
$0.52 < h \le 0.65$	1.17	
$0.65 < h \le 0.75$	1.22	
$0.75 < h \le 0.85$	1.28	
$0.85 < h \le 0.95$	1.35	
$0.95 < h \le 1.05$	1.414	$Max(\lvert\Delta p\rvert$ or $\lvert\Delta q\rvert) \times K$
$1.05 < h \le 1.15$	1.35	
$1.15 < h \le 1.35$	1.28	
$1.35 < h \le 1.50$	1.22	
$1.50 < h \le 1.95$	1.17	
$1.95 < h \le 3.50$	1.08	
$3.50 < h$	1.00	

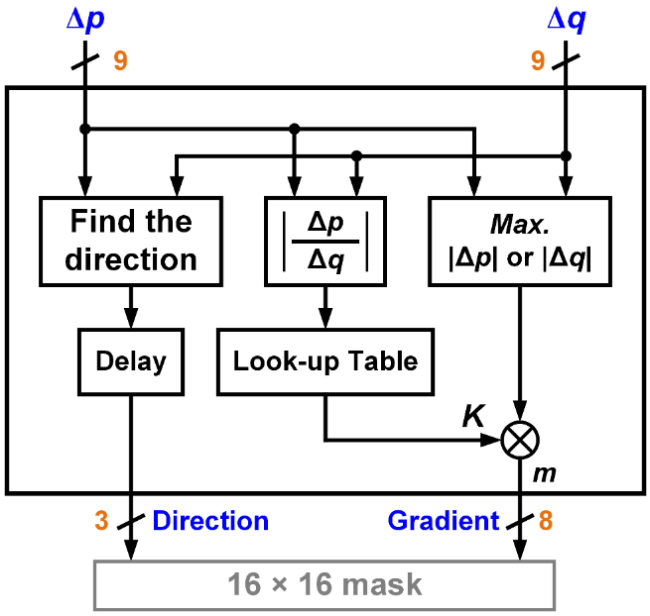


Figure 7. Function blocks of the proposed TI and look-up table approach for determining the direction and gradient of the detection point.

2.2.2. Feature Descriptor

The feature descriptor consists of the gradient sums in four directions for each 4 × 4 partition area in a 16 × 16 mask, as shown in Figure 8a. Since one 16 × 16 mask area can contain sixteen 4 × 4 partitions, the feature descriptor with a four-direction representation will possess 64 dimensions, which are half of that by conventional methods, as shown in Figure 8b. It not only reduces the required devices in implementation but also accelerates the producing speed of feature descriptors. These benefits are also helpful to the subsequent feature matching.

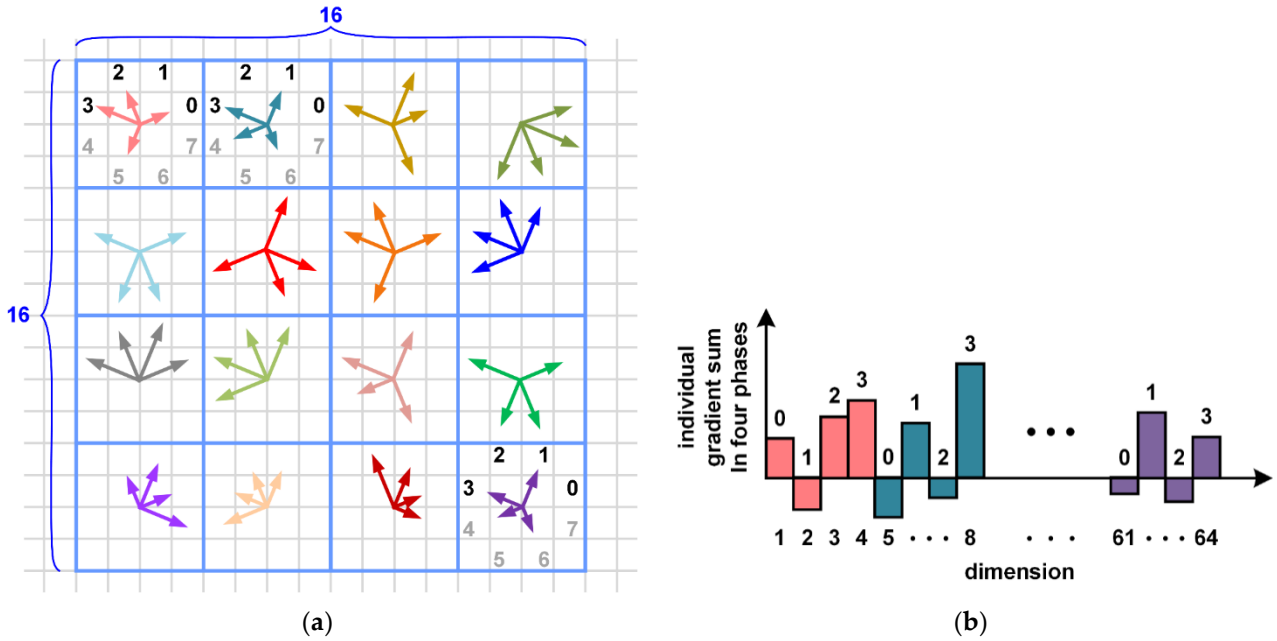


Figure 8. (**a**) Sixteen 4 × 4 partitions in a 16 × 16 mask for the feature descriptor and (**b**) the corresponding gradient sum in each 4 × 4 partitions.

2.2.3. Normalization

For general images, large differences usually exist among the gradient sums, resulting in different dimensions of the feature descriptor. Consequently, the feature descriptor, $OF = (o_1, o_2, \ldots, o_{64})$, needs to be normalized to decrease the scale without significantly altering the distribution among gradient sums. The normalization of the feature descriptor is achieved in this paper by:

$$F = (f_1, f_2, \cdots, f_{64}) = R \times \frac{OF}{W} \tag{6}$$

where W is the sum of all gradient sums in all dimensions of the feature descriptor.

$$W = \sum_{i=1}^{64} o_i \tag{7}$$

A scaling factor, $R = 255$, is chosen here for better discrimination. That is, the resulting gradient sums of the feature descriptor lie between ±255. Thus, the required bit number of the gradient sum is reduced from 13 bits to 9 bits in the normalization of the feature descriptor. In the implementation of the presented normalization, we introduce an additional multiplier having the power of 2 approximating the sum of all gradients for achieving a simple operation. Thus, the normalized feature descriptor F can be easily obtained by only shifting bits of the feature descriptor. The realized block diagram of the normalization is shown in Figure 9.

$$F = \frac{R \times 2^j}{W} \times OF \times \frac{1}{2^j} \tag{8}$$

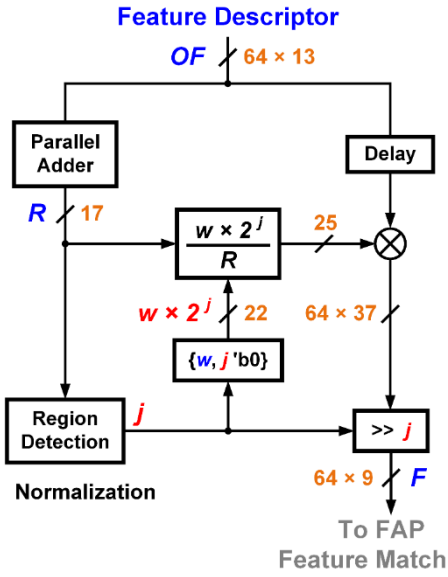


Figure 9. Function blocks of the normalization of the feature descriptor.

2.3. Feature Detection

In the beginning, we use three levels of DoG images to establish a 3D space. A 3 × 3 mask will then be applied to the three DoG images to define 27-pixel data for feature detection, as shown in Figure 10. Thus, center pixel of the 3 × 3 mask in the 2nd level DoG image, D2, is regarded as the detection point. Three kinds of detections, including extrema detection, high contrast detection, and corner detection, are adopted and processed in parallel, as shown in Figure 11.

Once the results of the three detections are all positive, the detection point becomes a feature point, and a detected signal will be sent to the feature descriptor and feature matching blocks for further processing.

To evaluate the variation of the pixel values around the detection point, 3D and 2D Hessian matrix, $H_{3\times3}$ and $H_{2\times2}$, respectively, will be used in the high contrast and corner detections.

$$H_{2\times2} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{9}$$

$$H_{3\times3} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}, \tag{10}$$

where the definitions of the elements are listed as follows.

$$D_{xx} \equiv D_2(x+1, y) + D_2(x-1, y) - 2D_2(x, y) \tag{11}$$

$$D_{yy} \equiv D_2(x, y+1) + D_2(x, y-1) - 2D_2(x, y) \tag{12}$$

$$D_{zz} \equiv D_3(x, y) + D_1(x, y) - 2D_2(x, y) \tag{13}$$

$$D_{xy} \equiv \frac{D_2(x+1, y+1) - D_2(x+1, y-1)}{4} - \frac{D_2(x-1, y+1) - D_2(x-1, y-1)}{4} \tag{14}$$

$$D_{xz} \equiv \frac{D_3(x+1, y) - D_1(x+1, y)}{4} - \frac{D_3(x-1, y) - D_1(x-1, y)}{4} \tag{15}$$

$$D_{yz} \equiv \frac{D_3(x, y+1) - D_1(x, y+1)}{4} - \frac{D_3(x, y-1) - D_1(x, y-1)}{4} \tag{16}$$

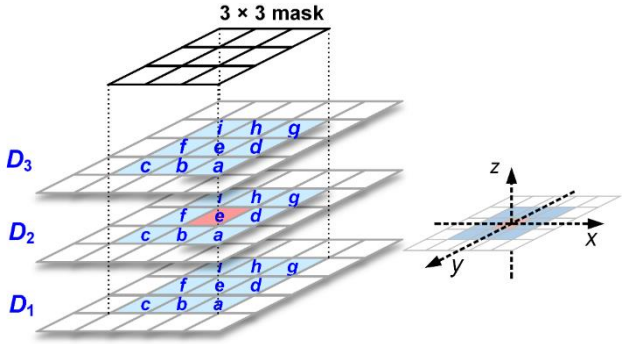


Figure 10. The detection range includes 27 pixels in three DoG images under a 3 × 3 mask, where the orange pixel is regarded as the detection point.

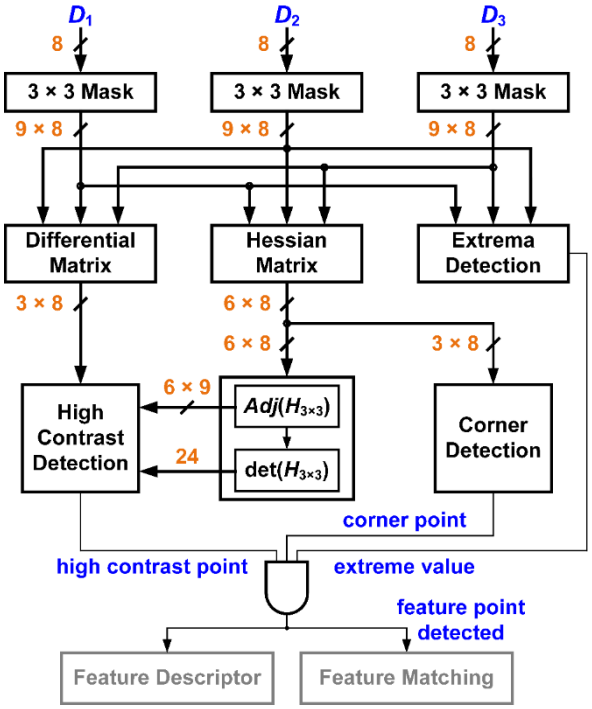


Figure 11. Building blocks of the feature detection.

Since both the matrices are symmetric, only six elements need to be determined. Figure 12 shows the realized structure, where pixel inputs come from the corresponding positions shown in Figure 10.

2.3.1. Extreme Detection

In extreme detection, the detection point is compared with the other 26 pixels in the 3D mask range. If the pixel value of the detection point is the maximum or the minimum one, the detection point will be a possible feature point [18].

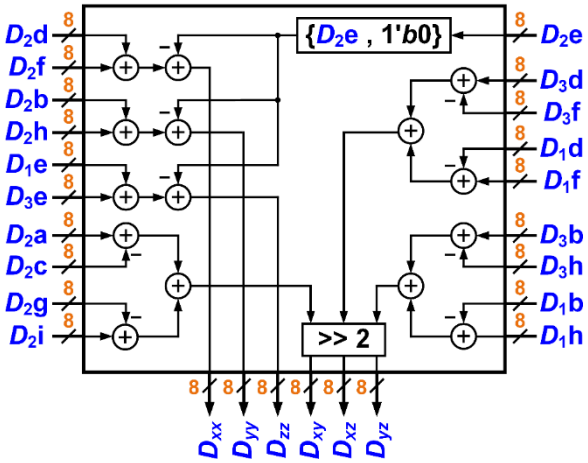


Figure 12. Hardware structure of the elements in 2D and 3D Hessian matrices.

2.3.2. High Contrast Detection

In conventional approaches, when the variation of the pixel values in three-dimension space around the detection point is greater than a threshold of 0.03, the high contrast feature can be assured [11]. Here, to simplify the derivations and subsequent hardware design, an approximated value of 1/32 is chosen to be the threshold value. That is, the resulting pixel variation criterion is modified as:

$$|D(S)| > \frac{1}{32}, \tag{17}$$

where $S = [x\ y\ z]^T$ denotes the 3D coordinate vector. The pixel variation can be expressed as the Maclaurin series and approximated by:

$$D(S) \approx D(0) + \frac{\partial D^T(0)}{\partial S} \cdot S + \frac{1}{2} S^T \cdot \frac{\partial^2 D(0)}{\partial S^2} \cdot S \tag{18}$$

where $D(0)$ represents the pixel value of the detection point. The transpose of the partial derivative of $D(0)$ with respect to S will be:

$$\left[\frac{\partial D(0)}{\partial S}\right]^T = \frac{\partial D^T(0)}{\partial S} = [D_x \quad D_y \quad D_z], \tag{19}$$

where

$$D_x \equiv \frac{D_2(x+1, y) - D_2(x-1, y)}{(x+1)-(x-1)} = \frac{D_2\mathrm{d} - D_2\mathrm{f}}{2} \tag{20}$$

$$D_y \equiv \frac{D_2(x, y+1) - D_2(x, y-1)}{(y+1)-(y-1)} = \frac{D_2\mathrm{b} - D_2\mathrm{h}}{2} \tag{21}$$

$$\text{and} \quad D_z \equiv \frac{D_3(x, y) - D_1(x, y)}{3-1} = \frac{D_3\mathrm{e} - D_1\mathrm{e}}{2}. \tag{22}$$

Figure 13 shows the realized hardware of the partial derivative. The second partial derivative of $D(0)$ is the same as the 3D Hessian matrix.

$$\frac{\partial^2 D(0)}{\partial S^2} = H_{3\times 3} \tag{23}$$

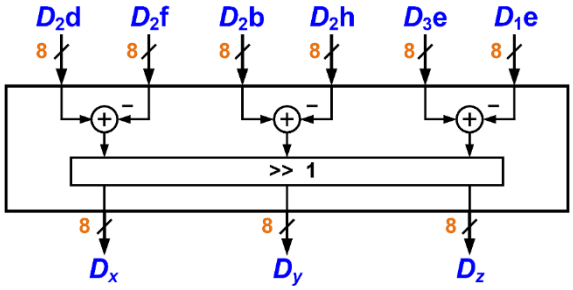


Figure 13. Hardware architecture of partial derivative of $D(0)$.

Let the first derivative of Equation (18) with respect to S equals zero.

$$D'(S) = 0 \tag{24}$$

The most variation of the pixel values around the detection point can be found at:

$$S = -\left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S} \tag{25}$$

Substituting Equation (25) into Equation (18), we obtain:

$$D(S) = D(0) - \frac{\partial D^T(0)}{\partial S} \cdot \left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S} + \frac{1}{2}\left[\left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S}\right]^T \cdot H_{3\times3} \cdot \left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S}$$

$$= D(0) - \frac{1}{2}\frac{\partial D^T(0)}{\partial S} \cdot \left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S}. \tag{26}$$

Substituting Equation (26) into Equation (17), we have:

$$\left| D(0) - \frac{1}{2}\frac{\partial D^T(0)}{\partial S} \cdot \left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S} \right| > \frac{1}{32}$$

$$\Rightarrow 16 \times \left| 2D(0) - \frac{\partial D^T(0)}{\partial S} \cdot \left(H_{3\times3}\right)^{-1} \cdot \frac{\partial D(0)}{\partial S} \right| > 1 \tag{27}$$

Since the matrix inversion, $(H_{3\times3})^{-1}$, requires lots of dividers in implementation, the resulting hardware cost would be high. Thus, the adjugate matrix, $Adj(H_{3\times3})$, and the determinant, $\det(H_{3\times3})$, are utilized here to accomplish the matrix inversion instead of using dividers.

$$\left(H_{3\times3}\right)^{-1} = \frac{Adj\left(H_{3\times3}\right)}{\det\left(H_{3\times3}\right)}, \tag{28}$$

where

$$Adj(H_{3\times3}) = \begin{bmatrix} +\begin{vmatrix} D_{yy} & D_{yz} \\ D_{yz} & D_{zz} \end{vmatrix} & -\begin{vmatrix} D_{xy} & D_{xz} \\ D_{yz} & D_{zz} \end{vmatrix} & +\begin{vmatrix} D_{xy} & D_{xs} \\ D_{yy} & D_{yz} \end{vmatrix} \\ -\begin{vmatrix} D_{xy} & D_{yz} \\ D_{xz} & D_{zz} \end{vmatrix} & +\begin{vmatrix} D_{xx} & D_{xz} \\ D_{xz} & D_{zz} \end{vmatrix} & -\begin{vmatrix} D_{xx} & D_{xz} \\ D_{xy} & D_{yz} \end{vmatrix} \\ +\begin{vmatrix} D_{xy} & D_{yy} \\ D_{xz} & D_{yz} \end{vmatrix} & -\begin{vmatrix} D_{xx} & D_{xy} \\ D_{xz} & D_{yz} \end{vmatrix} & +\begin{vmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{vmatrix} \end{bmatrix} \tag{29}$$

and

$$\det\left(H_{3\times3}\right) = \begin{vmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{vmatrix} \tag{30}$$

Figure 14 shows the realized architecture of the determinant and adjugate matrices. Since the adjugate matrix is symmetric, only six elements need to be processed in the implementation. The determinant of $H_{3\times3}$ can be obtained by repeatedly using the first row of the adjugate matrix combined with elements in the first column of the determinant to reduce the hardware cost. Thus, Equation (27) can be rewritten as:

$$16\times\left|2D(0)-\frac{\partial D^T(0)}{\partial S}\cdot\frac{Adj(H_{3\times3})}{\det(H_{3\times3})}\cdot\frac{\partial D(0)}{\partial S}\right|>1$$
$$\Rightarrow 16\times\left|2D(0)\cdot\det\left(H_{3\times3}\right)-Mul\right|>\left|\det\left(H_{3\times3}\right)\right|, \tag{31}$$

where

$$\Rightarrow Mul=\frac{\partial D^T(0)}{\partial S}\cdot Adj\left(H_{3\times3}\right)\cdot\frac{\partial D(0)}{\partial S} \tag{32}$$

That is, the detection point will have a high-contrast feature and could be regarded as a possible feature point when Equation (31) is satisfied. Since the condition for the high-contrast feature detection is less complicated than the conventional ones, the resulting hardware circuits will be simpler for achieving a higher operating speed [12].

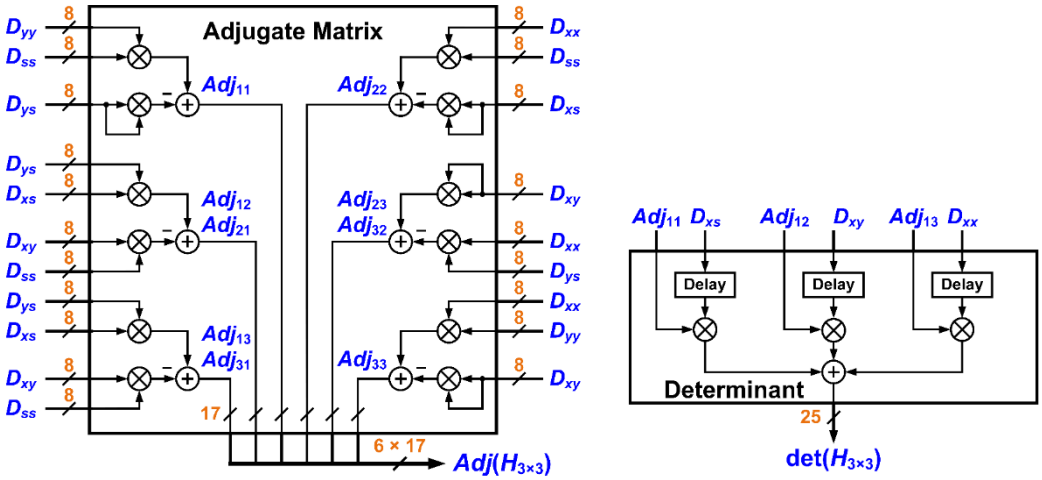


Figure 14. Hardware architecture of the determinant and adjugate matrices.

2.3.3. Corner Detection

In this paper, the Harris corner detection is adopted [19],

$$r\times tr^2(H_{2\times2})>(r+1)^2\cdot\det(H_{2\times2}) \tag{33}$$

where the parameter r is the ratio of two eigenvalues of $H_{2\times2}$ and $r > 1$, and $tr(H_{2\times2})$ and $\det(H_{2\times2})$ respectively represent the trace and the determinant of $H_{2\times2}$, which are given by:

$$tr\left(H_{2\times2}\right)=D_{xx}+D_{yy} \tag{34}$$

and $$\det\left(H_{2\times2}\right)=\begin{vmatrix}D_{xx} & D_{xy}\\ D_{xy} & D_{yy}\end{vmatrix}=D_{xx}D_{yy}-D_{xy}^2 \tag{35}$$

That is, the detection point has a corner feature if the inequality Equation (33) is satisfied. An empirical value, $r = 10$, is chosen here for acquiring a better result. Figure 15 shows the hardware architecture of the corner detection.

Once the results of the three detections are all positive, the detection point will be logically regarded as a feature point. During the serial input of pixels, the production of the feature descriptor and the feature point detection proceeds until no image pixel remains.

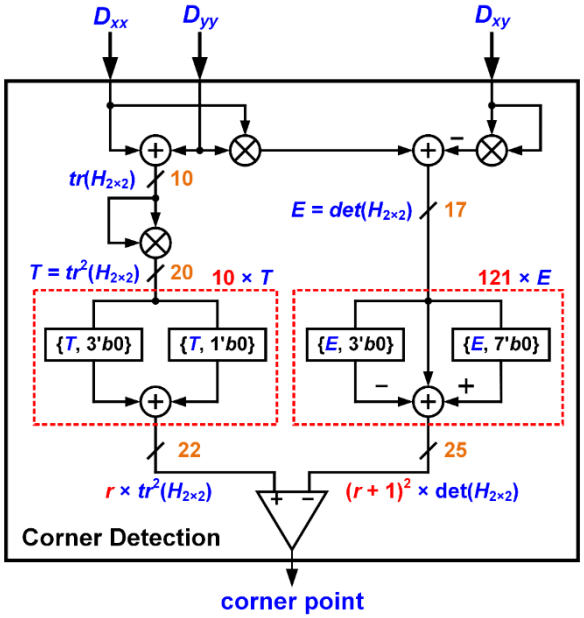


Figure 15. Hardware structure of the corner detection.

3. Finite-Area Parallel Matching for Stereo Vision

A double E-SIFT with FAP feature matching system is proposed for use in stereo vision, as shown in Figure 16. The stereo vision captures images by parallel cameras for the left and right E-SIFTs respectively to find the feature points and the corresponding feature descriptors, FL and FR. At the same moment, the corresponding coordinates will be output by a counter triggered by the detected signals, dtc_L and dtc_R, from the two E-SIFTs. Then, the stereo matching points, MX_L, MY_L, MX_R, and MY_R, will be found out and recorded by the proposed FAP feature matching. Since the image transmission could be interrupted due to the finite bandwidth of SDRAM or noise coupling, a data valid signal is added to alarm all building blocks to prevent abnormal termination or any discontinuous situation from corrupting correct output messages.

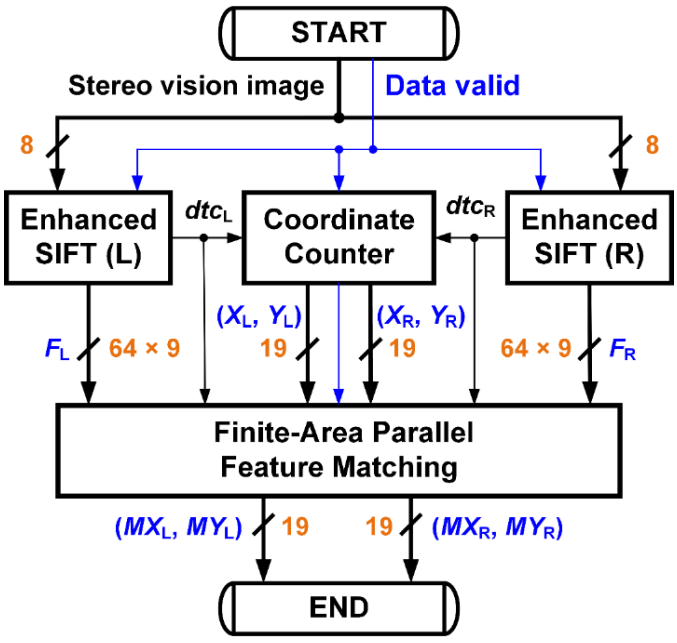
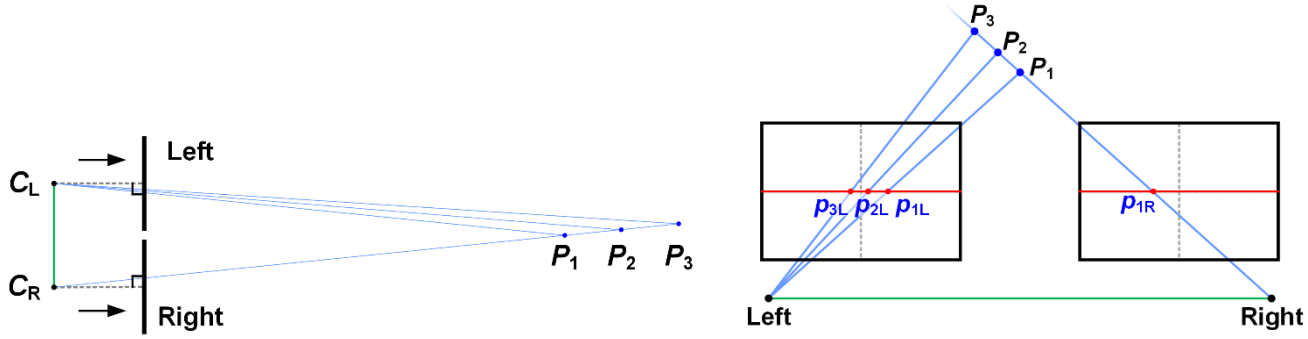


Figure 16. Building blocks of the proposed E-SIFT and FAP feature matching for stereo vision.

3.1. Finite-Area Parallel Feature Matching

Suppose two cameras, C_L and C_R, are set up in parallel at the same height without any tilt angle difference, and three objects, P_1, P_2, P_3, are placed in line with the right camera, as shown in the top view in Figure 17a [20]. In theory, the projections of the same object onto two image planes will be on the same row. However, based on Epipolar geometry [15], the imaging of P_2 and P_3 in the right image could disappear due to the shading effect caused by P_1. As shown in Figure 17b, three imagings, P_{1L}, P_{2L}, and P_{3L}, appear in the left image, but only one projection, P_{1R}, in the right one. Therefore, to avoid the feature points of the left and right images from error matching, a minimum threshold value for comparison will be chosen in the proposed FAP feature matching.



(**a**) (**b**)

Figure 17. (a) Top view and (b) front view of stereo vision with Epipolar phenomenon.

In reality, the object projections on two cameras cannot be on the same row due to a slight difference in the tilt angle between the two cameras. The resulting feature point matching would be inaccurate. However, too many feature points would consume a large number of resources. On the other hand, matching accuracy will decrease if there are not sufficient feature points.

3.2. Hardware Design

3.2.1. Parallel Arrangement for Feature Data

In stereo vision, the scenery of the left camera is shifted toward the left slightly compared to the right camera. Some objects would not be captured in the right camera and vice versa, as shown in the pink and blue areas of Figure 18. Thus, while pixels, which begin from the top left of the image, are entered into the system, mismatch could occur due to the visual angle difference between two the cameras. Therefore, a little delay will be added in the right-image input path to ensure the feature points found from the left image can be matched with the ones from the right image, as shown in the green area of the figure.

Figure 19 shows the building blocks of the proposed FAP feature matching. A demultiplexer transfers the serial feature point data into a parallel form by the left detected signal, dtc_L. The feature descriptors and the corresponding coordinates will be stored into registers. Then, sixteen feature points data of the left image will be compared with the feature point of the right image simultaneously when the right detected signal, dtc_R, is received.

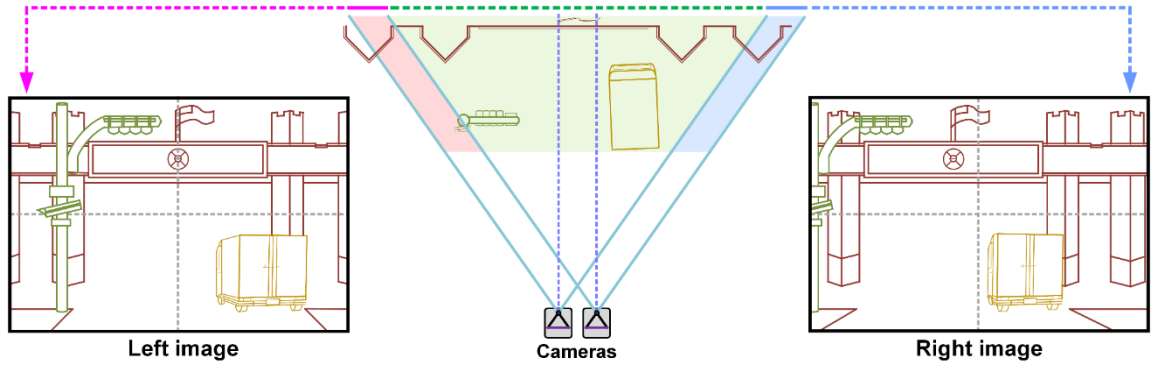


Figure 18. Different scenes are captured in stereo vision due to the visual angle difference between two cameras.

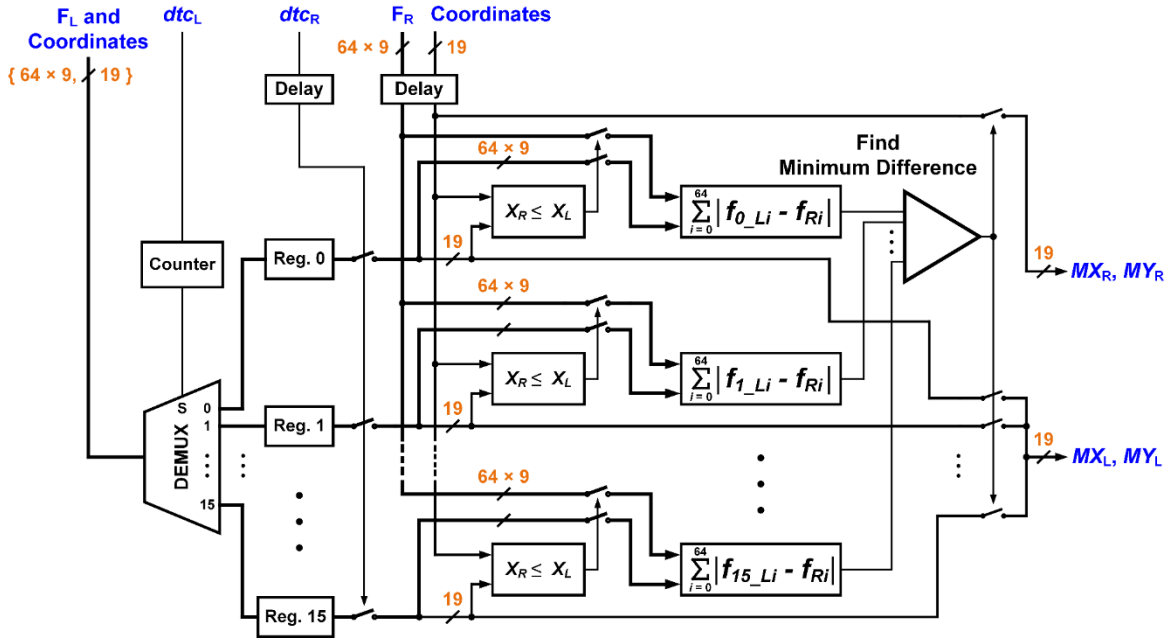


Figure 19. Building blocks of the proposed finite-area parallel (FAP) feature matching for stereo vision.

3.2.2. Feature Matching Algorithm

For a street view image of 640 × 480 pixels, about 2500 feature points can be detected. Every row of the image may include 5.2 feature points in average. Thus, to improve the accuracy of feature matching, 16 feature points, which are equivalent to an area of approximately three rows of pixels, will be chosen for feature matching in this paper.

In general, for the same object, the projection in the left image will be located in a little more to the right position than the corresponding one in the right. An exception is that if the object is far away from the cameras, the resulting x coordinates of the projections on two images will be almost equal. Thus, the comparison will proceed only the following condition is satisfied:

$$x_R \le x_L, \tag{36}$$

where x_R and x_L represent the right-image and left-image x-coordinates, respectively. To quickly find out the matching points, the 16 feature descriptors of the left image will be simultaneously subtracted from the feature descriptor of the right one. The resulting differences in 64 dimensions between each two feature descriptors are then summed together for comparison.

$$\eta_k = \sum_{i=1}^{64} \left| f_{k\_Li} - f_{Ri} \right|, \quad k = 0,\ 1,\ 2,\ \ldots,\ 15, \tag{37}$$

where f_{k_Li}, i = 1, 2, …, 64, is the element of the k-th feature descriptor in the left image, and f_{Ri} is the element of the feature descriptor of the right one. If the minimum sum is smaller than the half of the second minimum one, the feature points causing the minimum sum will be the matching pairs. The corresponding coordinates, MX_R, MY_R, MX_L, and MY_L, on the respective images will be the output.

4. Experiment Results

In this paper, we use Altera development board, DE2i-150, to verify the feasibility of the proposed E-SIFT and FAP feature matching. The core of DE2i-150 is Cyclone IV GX: EP4CGX150DF31C7 and the system clock is 50 MHz. Three types of experiments are conducted in the following. In the first one, the same functions are synthesized for two distinct SIFTs by the proposed TI with LUT approach and the CORDIC. Then, a comparison between the two SIFTs proceeds after measurements. In the second one, the performance of the proposed E-SIFT is evaluated and compared with other hardware-implemented SIFTs. In the final one, the system of the proposed double E-SIFT with FAP feature matching for stereo vision is evaluated and compared with the state-of-the-art approaches.

4.1. Proposed TI with LUT and CORDIC Algorithm

In the beginning, the proposed TI with LUT and the CORDIC algorithms are respectively realized into two distinct SIFTs. Next, we arbitrarily choose seven 640 × 480 images having different sceneries as samples for testing to acquire an objective evaluation from the comparison of the two distinct SIFTs.

In each experiment, both the original and skewed images are applied to the same SIFT to detect the feature points. Then, the matching pairs are determined by a conventional exhaustion method via C++ and Visual Studio. Each of the feature points on the original image will be compared with all feature points on the skewed ones to find the matching pairs. Based on the matching results, the resulting accuracy can be estimated.

Table 3 shows the performance and hardware resources required by the two approaches. Note that the accuracy between the proposed TI with LUT and CORDIC is very close. However, their consumed hardware sources are quite different. The quantities of LE and register in the proposed TI with the LUT approach are significantly reduced by 89.18% and 96.08% compared with the CORDIC algorithm.

Table 3. Accuracy and hardware comparison for proposed TI with LUT and CORDIC algorithm.

Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7
						

Image	Accuracy (%)							LEs	Reg.	DSP/Mult.	RAM (kbits)
	1	2	3	4	5	6	7				
CORDIC [12]	69.23	93.33	100.0	78.37	88.09	96.0	88.64	934	562	0	0.03
Proposed TI with LUT	70.37	94.44	100.0	76.31	86.66	95.65	91.30	101	22	0	0

4.2. Performance of Proposed E-SIFT

To verify the performance of the proposed E-SIFT, a 640 × 480 image having a static object, as shown in Figure 20a, is processed by a single E-SIFT to extract the feature points indicated by red points in Figure 20b, where 283 feature points are detected. Then, the image is shifted toward the bottom right direction for matching with the original one by a software-based feature matching, as shown in Figure 21. There are 290 feature points detected in the shifted image, resulting in 83 matching pairs. No matching error occurred

in these matching pairs. The resulting accuracy of the feature matching based on the proposed E-SIFT is 100%.

When the original image is rotated and skewed, 272 and 235 feature points are derived, respectively, as shown in Figure 22a,b. Through the proposed E-SIFT and the software-based feature matching, 13 and 11 matching pairs are obtained between the original image and the rotated and skewed images, respectively. The resulting accuracy of the feature matching based on the proposed E-SIFT is 100%.

To test the flexibility of the proposed E-SIFT, outdoor and indoor images are also skewed for feature matching with the original one, as shown in Figure 23a,b, respectively, where the matching accuracy reaches 88.89% and 92.39%, respectively, for these two images. The experimental results have shown that the proposed E-SIFT exhibits good performance even for images having cluttering scenery.

(**a**) (**b**)

Figure 20. (**a**) A 640 × 480 image for evaluating the proposed E-SIFT; (**b**) feature points detected.

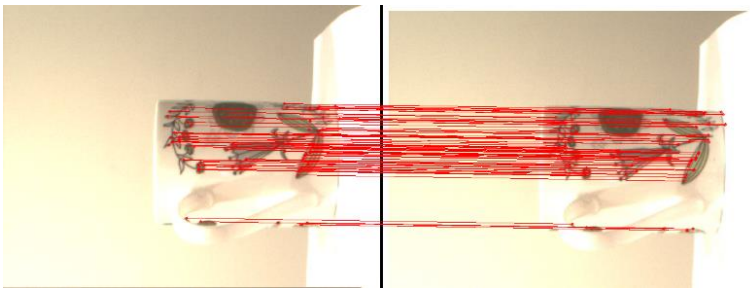
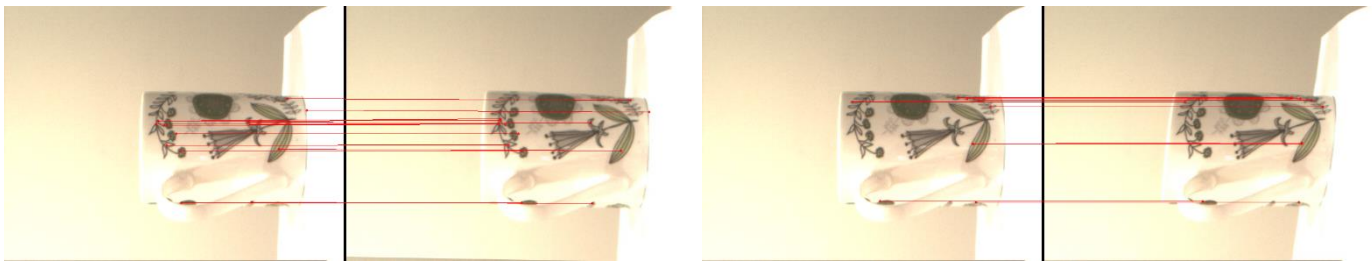


Figure 21. Feature matching results of the original image with the shifted one.



(**a**) (**b**)

Figure 22. Feature matching results of the original image with (**a**) the rotated and (**b**) the skewed image.

(**a**) (**b**)

Figure 23. (**a**) Matching accuracy for a skewed outdoor image is 92.39%, where 85 out of 92 feature points are successfully matched; (**b**) Matching accuracy for a skewed indoor image is 88.89%, where 48 out of 54 feature points are successfully matched.

Table 4 shows a performance comparison and hardware resources required by the proposed E-SIFT and other published papers. The processing time of one image, including detection of the feature points and generation of the feature descriptor, is 4.865 ms, reaching a corresponding frame rate of 205 fps. As shown in Table 4, the proposed E-SIFT is the fastest among the hardware-implemented SIFTs published in the literature. The resources required for the proposed E-SIFT includes 54,911 LEs, 36,153 registers, 297 DSP, and 267.9 kbits RAM. Although the SIFT in [11] utilized a similar quantity of devices, the resulting frame rate is 58 fps only. Compared with other published SIFTs, the proposed E-SIFT exhibits a faster response while using a relatively small quantity of hardware devices.

Table 4. Performance comparison of proposed E-SIFT with other published papers.

Mono SIFT	Resolution	Operating Frequency	Devices	Frame Rate (fps)	LEs	Registers	DSP/Multi.	RAM (kbits)
Proposed E-SIFT	640 × 480	50 MHz	Altera Cyclone IV GX	>205	54,911	36,153	297	267.9
[10]		21.7 MHz	Altera Cyclone IV GX	≈70	125,644	8,372	77	406
[11]		170 MHz	Xilinx Virtex-6	≈58	60,837	34,166	377	N/A
[12]		50 MHz	Altera Cyclone IV GX	<150	65,560	39,482	642	319
[16]		100 MHz	TSMC 0.18μm CMOS	30	1,320,000	N/A	N/A	5729
[21]		190 MHz	DC Ultra 130nm	20	548,000	N/A	N/A	448.7
[22]		40.355 MHz	Xilinx Zynq-7020	131.36	47,255	42,267	136	128
[23]		N/A	Xilinx Virtex-5	30	57,598	24,988	8	1206

4.3. Proposed Double E-SIFTs with FAP Feature Matching for Stereo Vision

In this paper, a stereo vison architecture of double E-SIFT integrated with FAP feature matching is proposed, as shown in Figure 24, where stereo images from the KITTI dataset are firstly written into SDRAM of Altera DE2i-150 by Nios II. When the system is activated, the stereo images in the SDRAM will be delivered to the proposed stereo vision system through the Avalon bus. An end signal and coordinates of the matching pairs will be sent back to a PC through buffers after finding the feature points. In the following experiments, we use 1226 × 370 images from the KITTI dataset for the proposed double E-SIFT and FAP feature matching system [24]. To be able to compare with other published papers, the KITTI stereo images are tailored symmetrically with respect to the center to form an image having 640 × 370 pixels, as shown in Figure 25.

As shown in Figure 26a, through the proposed double E-SIFT, 1748 and 1664 feature points are detected in the left and right images 05 000000.png from the KITTI dataset, respectively. After the proposed FAP feature matchings, 497 matching pairs are obtained.

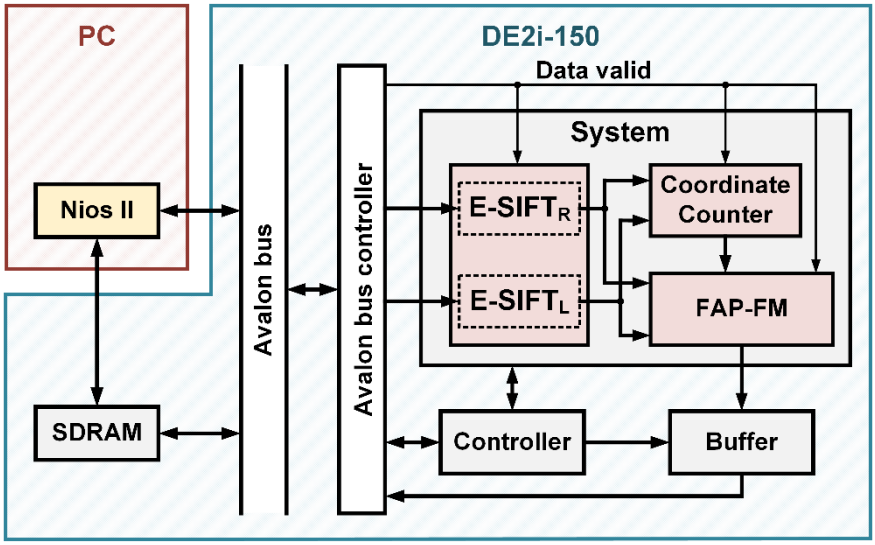


Figure 24. Architecture of the proposed stereo vison system.

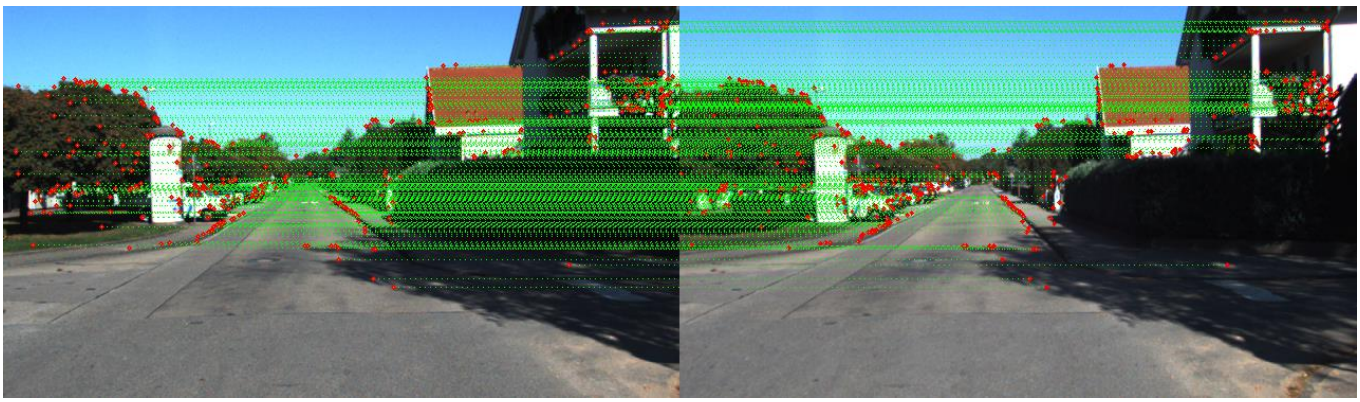
Figure 25. Image tailored from KITTI dataset.

When stereo images 00 02566.png from KITTI dataset having a car in the street are applied respectively to the left and right E-SIFTs, more feature points, i.e., 2162 and 1972 in the left and right images, can be detected, as shown in Figure 26b. Note that there are 102 and 110 feature points around the car in the left and right images, respectively. After the FAP feature matching, we obtain 560 matching pairs where 44 are for the car. Both the above-mentioned experiments of stereo visions present sufficient matching points for the purpose of mapping or object tracking. To evaluate the accuracy of the proposed double E-SIFT with FAP feature matching, the stereo images are aligned vertically, where green matching lines are depicted with the same procedure, as shown in Figure 27a,b. The accuracy of the proposed double E-SIFT with FAP feature matching presents good results because there are no apparent skewed matching lines existed in the images.

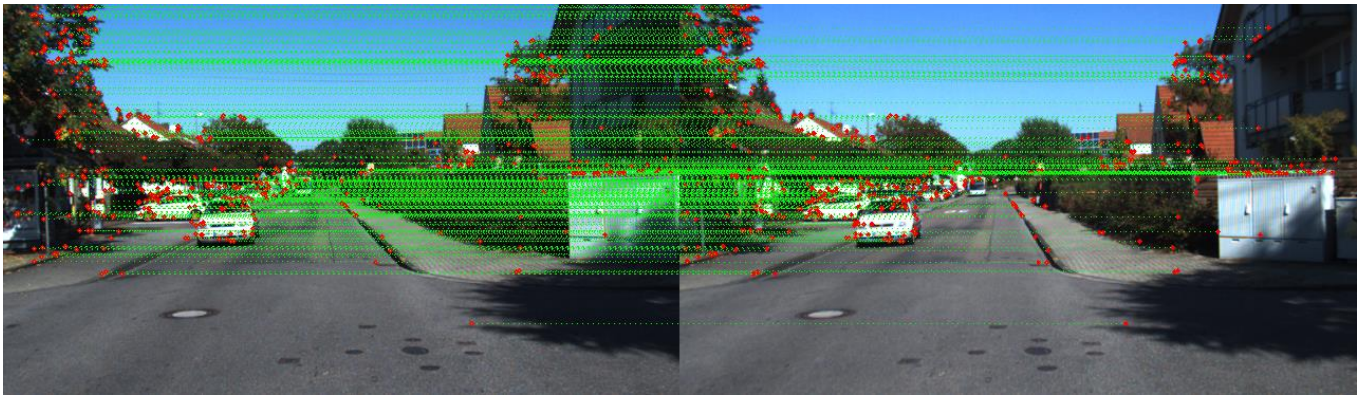
Table 5 shows the consumed hardware resources of the proposed double E-SIFT with FAP feature matching for stereo vision. The usage ratios of LEs, registers, RAM, and DSP/Multiplier in the DE2i-150 development board are 93.6%, 52.4%, 82.5%, and 8.9%, respectively. For the whole system, the double E-SIFT and the FAP feature matching occupy 76% and 24% of LEs, respectively, as shown in Figure 28a. Most of the registers, about 90%, are used by the E-SIFTs. Only 10% of the total register devices are used by the FAP feature matching, as shown in Figure 28b. The usage ratio of the coordinate counter is very low and can be neglected.

The total power consumption of the FPGA when it hosts the proposed architecture including the double E-SIFT and the FAP feature matching is less than 1168.33 mW. This value is estimated using the Power Analyzer from Intel [25]. It is difficult to directly measure the power consumption because the DE2i-150 development board that we used to

conduct the experiments does not come with a circuit for power estimation similar to the one used in [26].



(**a**)



(b)

Figure 26. (**a**) Stereo vision images, 05 000000.png from KITTI dataset, are applied to the proposed double E-SIFT with FAP feature matching, resulting in 497 matching pairs between the left and right images; (**b**) Stereo vision images, 00 02566.png from KITTI dataset, are applied to the proposed double E-SIFT with FAP feature matching, resulting in 560 matching pairs between the left and right images.

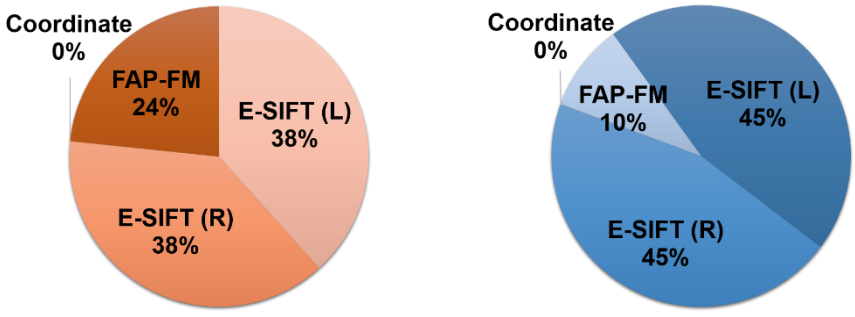
Table 6 lists the performance comparison and hardware resources usage of the proposed system with other published papers. The resulting frame rate of the proposed double E-SIFT with FAP feature matching is 181 fps, which is faster than most of the approaches. Although reference [27] implementing SURF detector and BRIEF descriptor has the fastest frame rate [27], the required hardware resource is much higher than the proposed method in this paper. The quantities of LEs in [14] and [28] are close to the proposed stereo vision system, but the resulting frame rates are only about 40 fps. The stereo vision system proposed by [29] uses many hardware resources, but the resulting frame rate is only half of the proposed approach. In summary, the proposed double E-SIFT with FAP feature matching has a satisfactory frame rate with an acceptable hardware cost.

Table 5. Hardware resources utilized in the proposed system.

Module Name	LEs	Registers	DSP/Multi.	RAM (kbits)
E-SIFTs for Stereo vision	107,481 (71.7%)	71,182 (47.5%)	594 (82.5%)	550.8 (8.3%)
Coordinate Counter	84 (0.006%)	39 (0.003%)	0 (0.0%)	0 (0.0%)
FAP Feature Matching	32,738 (21.8%)	7368 (4.9%)	0 (0.0%)	35.6 (0.54%)
Overall structure	**140,303 (93.6%)**	**78,589 (52.4%)**	**594 (82.5%)**	**588.8 (8.9%)**

(**a**) (b)

Figure 27. The same stereo visions as (**a**) Figure 26a, and (**b**) Figure 26b are aligned vertically to verify the accuracy. There is no apparent skewed matching line existed in the images.



(**a**) (**b**)

Figure 28. Usage ratio of (**a**) LE used and (**b**) registers used for the proposed E-SIFTs and FAP feature matching.

Table 6. Performance comparison of proposed stereo E-SIFTs and FAP feature matching with other published papers.

	Resolution	Operating Frequency	Devices	Approach	Frame Rate (fps)	LEs	Registers	DSP/Multi.	RAM (kbits)
Proposed Architecture	640 × 370	50 MHz	Altera Cyclone IV GX	E-SIFT × 2 + FAP-FM	>180	140,303	78,589	594	588.8
[14]	640 × 480	12.5 MHz [1] 15 MHz [2]	Altera Cyclone IV	SIFT + FM + R	40	141,394	25,076	528	1743
[27]	512 × 512	1070 MHz	Xilinx Virtex-7	SURF + BRIEF	380	267,095	298,864	144	11
[28]	640 × 480	12.5 MHz	Altera Cyclone IV GX	SIFT + FM	>36	138,944	24,079	169	2860
[29]	640 × 480	25 MHz	Altera Stratix IV	SIFT + FM + R	>81	494,201	105,423	960	1886

FM: Feature Matching; R: Random sample consensus (RANSAC). SURF: Speed Up Robust Features; BRIEF: Binary Robust Independent Elementary Features. [1] Frequency of SIFT and FM; [2] Frequency of RANSAC.

5. Conclusions

In this paper, we propose an FPGA-implemented double E-SIFT with FAP feature matching for stereo vision. With the improved architecture for the Gaussian pyramid, the Gaussian-blurred image and DoG pyramid can be produced simultaneously in only one clock cycle. As for the feature descriptor, we propose a simple TI with LUT method to simplify the decision of direction and gradient of detection points. The dimension of the feature descriptor is also decreased by half so that the hardware cost is significantly reduced. In the feature decision part, in comparison with the conventional approaches, a simple condition for high-contrast detection is derived by approximating the threshold value to the power of two. Thanks to the proposed double E-SIFT integrated with FAP feature matching, matching pairs between two images can be efficiently determined. Based on the position of the feature point in the right image, the corresponding area in the left image can be chosen for feature matching without external memory.

Since the simplification of the structures and approaches adopted in the proposed E-SIFT, the quantity of hardware devices required in the implementation can be significantly reduced than that by other published papers. It also effectively improves the operating speed of the proposed system. From the experimental results, a frame rate of 205 fps can be reached by the proposed E-SIFT at a system clock of 50 MHz. For stereo vision from the KITTI dataset, a frame rate of 181 fps is also achieved by the proposed double E-SIFT with FAP feature matching system. Besides, a data valid signal is added in the system to synchronize all the functional blocks to prevent the proposed circuitries from ceasing transmission due to the finite bandwidth of SDRAM or the noise interference.

Author Contributions: Conceptualization, C.-H.K. and C.-C.H.; implementation, E.-H.H.; validation, C.-H.K. and C.-H.C.; writing—original draft preparation, E.-H.H. and C.-H.C.; writing—review and editing; C.-H.K. and C.-C.H.; visualization, E.-H.H.; supervision, C.-H.K. and C.-C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the "Chinese Language and Technology Center" of National Taiwan Normal University (NTNU) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, and Ministry of Science and Technology, Taiwan, under Grants no. MOST 109-2634-F-003-006 and MOST 109-2634-F-003-007 through Pervasive Artificial Intelligence Research (PAIR) Labs.

Acknowledgments: We are grateful to the National Center for High-performance Computing for the computer time and facilities to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chien, C.H.; Hsu, C.C.; Wang, W.Y.; Chiang, H.H. Indirect Visual Simultaneous Localization and Mapping System Based on Linear Models. *IEEE Sens. J.* **2020**, *20*, 2738–2747.
2. Baltes, J.; Kung, D.W.; Wang, W.Y.; Hsu, C.C. Adaptive Computational SLAM Incorporating Strategies of Exploration and Path Planning. *Knowl. Eng. Rev.* **2019**, *34*, 1–18.
3. Zhao, B.; Huang, Y.; Wei, H.; Hu, X. Ego-Motion Estimation Using Convolutional Neural Networks through Optical Flow Learning. *Electronics* **2021**, *10*, 222.
4. Guo, R.; Peng, K.; Zhou, D.; Liu, Y. Robust Visual Compass Using Hybrid Features for Indoor Environments. *Electronics* **2019**, *8*, 220.
5. Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded Up Robust Features. In Proceedings of Computer Vision ECCV 2006, Graz, Austria, 7–13 May 2006; Volume 3951, pp. 404–417.
6. Nuari, R.; Utami, E.; Raharjo, S. Comparison of Scale Invariant Feature Transform and Speed Up Robust Feature for Image Forgery Detection Copy Move. In Proceedings of the 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 20–21 November 2019; pp. 107–112.
7. Li, Z.; Jia, H.; Zhang, Y.; Liu, S.; Li, S.; Wang, X.; Zhang, H. Efficient parallel optimizations of a high-performance SIFT on GPUs. *J. Parallel Distrib. Comput.* **2019**, *124*, 78–91.
8. Lin, C.H.; Wang, W.Y.; Liu, S.H.; Hsu, C.C.; Chien, C.H. Heterogeneous Implementation of a Novel Indirect Visual Odometry System. *IEEE Access* **2019**, *7*, 34631-34644.
9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
10. Vourvoulakis, J.; Kalomiros, J.; Lygouras, J. Fully pipelined FPGA-based architecture for real-time SIFT extraction. *Microprocess. Microsyst.* **2016**, *40*, 53–73.
11. Yum, J.; Lee, C.H.; Kim, J.S.; Lee, H.J. A Novel Hardware Architecture with Reduced Internal Memory for Real-Time Extraction of SIFT in an HD Video. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1943–1954.
12. Li, S.A.; Wang, W.Y.; Pan, W.Z.; Hsu, C.C.J.; Lu, C.K. FPGA-Based Hardware Design for Scale-Invariant Feature Transform. *IEEE Access* **2018**, *6*, 43850–43864.
13. Chien, C.H.; Chien, C.J.; Hsu, C.C. Hardware-Software Co-Design of an Image Feature Extraction and Matching Algorithm. In Proceedings of the 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 28 February–2 March 2019; pp. 37–41.
14. Vourvoulakis, J.; Kalomiros, J.; Lygouras, J. FPGA-based architecture of a real-time SIFT matcher and RANSAC algorithm for robotic vision applications. *Multimed. Tools Appl.* **2018**, *77*, 9393–9415.
15. Hartley, R.; Zisserman, A. Epipolar Geometry and the Fundamental Matrix. In *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004; pp. 239–261.
16. Huang, F.C.; Huang, S.Y.; Ker, J.W.; Chen, Y.C. High-Performance SIFT Hardware Accelerator for Real-Time Image Feature Extraction. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 340–351.
17. Volder, J.E. The CORDIC Trigonometric Computing Technique. *IRE Trans. Electron. Comput.* **1959**, *EC-8*, 330–334.
18. Bonato, V.; Marques, E.; Constantinides, G.A. A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1703–1712.
19. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.
20. Bertozzi, M.; Broggi, A. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans. Image Process.* **1998**, *7*, 62–81.
21. Yum, J.; Lee, C.H.; Park, J.; Kim, J.S.; Lee, H.J. A Hardware Architecture for the Affine-Invariant Extension of SIFT. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3251–3261.
22. Wilson, C.; Zicari, P.; Craciun, S.; Gauvin, P.; Carlisle, E.; George, A.; Lam, H. A Power-Efficient Real-Time Architecture for SURF Feature Extraction. In Proceedings of the 2014 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 8–10 December 2014; pp. 1–8.
23. Chiu, L.C.; Chang, T.S.; Chen, J.Y.; Chang, N.Y.C. Fast SIFT Design for Real-Time Visual Feature Extraction. *IEEE Trans. Image Process.* **2013**, *22*, 3158–3167.
24. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
25. Power Estimation and Power Analyzer. Available online: https://www.intel.com/content/www/us/en/programmable/support/support-resources/operation-and-testing/power/pow-powerplay.html (accessed on 29 June 2021).
26. Cardarilli, G.C.; Nunzio, L.D.; Fazzolari, R.; Re, M.; Silvestri, F.; Spanò, S. Energy consumption saving in embedded microprocessors using hardware accelerators. *Telkomnika Telecommun. Comput. Electron. Control* **2018**, *16*, 1019–1026.
27. Liu, D.; Zhou, G.; Zhang, D.; Zhou, X.; Li, C. Ground Control Point Automatic Extraction for Spaceborne Georeferencing Based on FPGA. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3350–3366.

28. Vourvoulakis, J.; Kalomiros, J.; Lygouras, J. A complete processor for SIFT feature matching in video sequences. In Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 21–23 September 2017; pp. 95–100.
29. Vourvoulakis, J.; Kalomiros, J.; Lygouras, J. FPGA accelerator for real-time SIFT matching with RANSAC support. *Microprocess. Microsyst.* **2017**, *49*, 105–116.