A 4.29nJ/pixel Stereo Depth Coprocessor With Pixel Level Pipeline and Region Optimized Semi-Global Matching for IoT Application

Pingcheng Dong, Zhuoyu Chen, Zhuoao Li, Yuzhe Fu, Lei Chen, and Fengwei An, *Member, IEEE*

***Abstract*— The semi-global matching (SGM) algorithm in stereo vision is a well-known depth-estimation method since it can generate dense and robust disparity maps. However, the real-time processing and low power dissipation, the specifications of the Internet-of-Thing (IoT) applications, are challenging for their computational complexity. In this paper, we propose a hardware-oriented SGM algorithm with pixel-level pipeline and region-optimized cost aggregation for high-speed processing and low hardware-resource usage. Firstly, the matching costs in a region are integrated with an optimization strategy to significantly reduce memory usage and improve the processing speed of the cost aggregation. Then, a two-layer parallel two-stage pipeline (TPTP) architecture, which enables pixel-level processing, is designed to calculate two directions (0° and 135°) aggregation to further solve the crucial computational bottleneck of the SGM algorithm. Finally, the architecture is demonstrated on a low-cost XILINX Spartan-7 device and an advanced Stratix-V FPGA device for VGA (640 × 480) depth estimation. The experimental results show that the proposed architecture with compact hardware architecture also ensures accuracy. The pixel-level pipeline architecture enables a processing speed of 355 frames per second (fps) at 109MHz on the Spartan-7 FPGA device and 508 fps at 156MHz on the Stratix-V FPGA. Besides, the coprocessor respectively achieves an energy efficiency of 4.74 nJ/pixel with a power dissipation of 517mW and 4.29nJ/pixel with a power dissipation of 669mW on these two FPGAs.**

***Index Terms*— Regional optimization, stereo vision, semi-global matching, real-time, FPGA.**

I. Introduction

BINOCULAR stereo vision is an important branch of computer vision [1], [2]. It is a technique to recover depth information from planar images by simulating the principle of human visual perception [3], [4]. Thus, the stereo matching technique has been widely used in a variety of application areas, including industrial production automation [5], mobile robot [6], self-driving cars (distance detection, navigation) [7], object detection [8], remote sensing image analysis, etc.

As for the depth estimation by stereo vision, the semi-global matching (SGM) [9] has been widely highlighted for its fast speed and robustness [10] in comparison to the global matching. However, the global matching algorithms consume enormous resources because their energy function is global. A typical SGM contains the matching cost computation, the cost aggregation, the disparity computation, the disparity refinements, and the depth transformation.

The Census Transform method for matching cost generation [11] is an image operator that associates a binary string to each pixel of a grayscale image for expressing the visual correspondence problems [12]. The cost aggregation essentially aims to optimize the matching costs of pixels in a few directions. The depth can be estimated by the winner-take-all (WTA) method to search minimum aggregated matching costs [13]. The complexity of the SGM mainly increases with disparity search range and cost aggregation paths. [14]. How to accelerate the SGM is an attractive research topic for real-time processing. Meanwhile, the Internet of Things (IoT) applications specified low power dissipation [15].

In this work, we propose a region-optimized SGM algorithm with pixel-level pipeline architecture for solving the computational bottleneck of the cost aggregation. Thus, the designed stereo-depth coprocessor can be applied in IoT edge devices with compact hardware architecture. The contribution of this work can be summarized as follows:

A) The region-optimized cost aggregation adopts a representative matching cost determined by an optimized strategy. In the case of the region size with four matching costs, the hardware-resource and memory usage can be saved over 84% comparing with other works, and the average accuracy loss is less than 1% in comparison to the original SGM. Furthermore, only the cost aggregation paths with the combination of 0° and 135° can achieve higher accuracy than four paths and eight paths.

B) A two-layer parallel two-stage pipeline (TPTP) architecture can parallel compute the cost aggregation of the 128-disparity search range. Hence, the pixel-level pipeline enables the simultaneous calculation of two directions aggregation and the speed synchronization

Manuscript received March 22, 2021; revised July 1, 2021, July 6, 2021, and July 17, 2021; accepted July 20, 2021. Date of publication August 3, 2021; date of current version January 10, 2022. This work was supported by the Shenzhen Science and Technology Innovation Commission under Grant JSGG20200102162401765. This article was recommended by Associate Editor D. John. *(Pingcheng Dong and Zhuoyu Chen contributed equally to this work.) (Corresponding authors: Fengwei An; Lei Chen.)*

Pingcheng Dong, Zhuoyu Chen, Zhuoao Li, Yuzhe Fu, and Lei Chen are with the School of Microelectronics, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: chenl03@pcl.ac.cn).

Fengwei An is with the Engineering Research Center of Integrated Circuits for Next-Generation Communications, Ministry of Education, School of Microelectronics, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: anfw@sustech.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3100071.

Digital Object Identifier 10.1109/TCSI.2021.3100071

1549-8328 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

to the stream of the image sensor. The proposed coprocessor can achieve a dynamic power dissipation of 63mW at 24MHz and 78 VGA (640 × 480)-frames per second (fps) for the IoT low-power scenarios. Meanwhile, the dynamic power dissipation of 517mW and a processing speed of 355 fps at the max working frequency of 109MHz on a low-cost XILINX Spartan-7 FPGA device are for high-speed application. In addition, the architecture implemented on the advanced Intel Stratix-V device can reach a maximum working frequency of 156MHz, a speed of 508fps with dynamic power dissipation of 669mW.

C) For satisfying the IoT applications, the developed architecture is so compact that it can be easily implemented on a low-cost XILINX Spartan-7 FPGA device. Moreover, the hardware-resource usage of the proposed work is lowest than the state-of-art works under the same disparity search range and image resolution.

The remains of this paper are organized as follows. Section II introduces the related works. Section III illustrates the proposed region-optimized SGM algorithm. Section IV elaborates on the implementation of the hardware architecture. Section V presents the experimental results with accuracy, hardware-resource usage, and performance. Finally, we conclude in Section VI.

II. Related Work

Accelerating stereo matching algorithms is very attractive in the past decades, but designing a compact and resource-friendly hardware architecture is still a challenge. Thus, accelerations based on CPUs, GPUs, FPGAs, and ASICs have been implemented to solve numerous critical issues, including high computational complexity, large storage requirement, extensive data access, and long processing latency.

Seki and Pollefeys [16] proposed an innovative convolutional neural network in the SGM algorithm on GPU, which provides learned penalties for every pixel. They implemented their system with Torch7 on NVIDIA Titan X and achieved ultra-high accuracy in the final depth map, with less than 3% Out-Noc error. However, 250W power dissipation is very difficult to apply IoT edge devices with GPUs. Additionally, Cambuim *et al.* [17] proposed a stereo matching system based on two distinct heterogeneous architectures (CPU and FPGA) and achieved a frame rate of 25 fps for the disparity maps processing XGA video (1024 × 768) with 256 disparity levels. However, the resource consumption was so high that it was only applicable to the advanced FPGA platforms with a large scale of logic elements.

As for standalone FPGA, in [18], an external DRAM was involved for the 128-disparity search range. They attained a processing speed of 324 fps and the power dissipation of 2.313W with the maximum frequency of 133MHz on the Virtex-5 FPGA platform. Jin *et al.* [19] proposed a fully pipelined stereo vision system with additional sub-pixel accuracy on Virtex-4 that could attain 230 fps with 93.1MHz and 64 disparity range for XGA (1024 × 768) video, but the resource consumption is high, which affects the power efficiency and hardware compactness. A depth estimation architecture based on guided image filtering was designed in [20] on the same FPGA platform to process 60 fps FHD (1920 × 1080) video. Meanwhile, on the advanced Virtex-7 FPGA [21], the power dissipation was only 172mW for XGA (1024 × 768) with a 64-disparity search range. A stereo vision system based on SGM with scalable resolutions and disparity search range was implemented on a low-cost FPGA platform, i.e., XILINX Spartan-7, for VGA (640 × 480) resolution with a 128-disparity search range. They attained a processing speed of 324 fps with a maximum frequency of 100 MHz [22]. Besides, in [23], image-guided depth inference, upsampling, and octave search range sampling were adopted for wide-depth-range scenes to save computation. The processor was implemented on XILINX ZC706 with a 128-disparity search range, 384mW, and 95pJ energy efficiency for 1920 × 1088 resolution, but the processing speed was 54MHz and the memory usage 329Kb is high. In [24], a non-iterative Patch Match and separable weighted median filtering algorithm was proposed to reduce the computational complexity of stereo matching and achieved 60 fps in a 128-disparity search range for FHD video on the Kintex-7 FPGA platform.

Regarding the solutions in Application Specific Integrated Circuit (ASIC), Chen *et al.* developed a tile-based belief propagation algorithm and utilized five views to improve the quality of data cost. The hardware architecture with a 64-disparity search range and 32 × 32 tiles in 40nm CMOS technology could reach 30 fps and 611mW for FHD resolution at 215MHz [25]. In [26], Lee *et al.* utilized a tile-based SGM architecture with a task-level pipeline which can reduce the external memory bandwidth by 85.5%. Furthermore, in [27], an 8 × 8 tile-based SGM processor with a 64-disparity search range was implemented in 65nm CMOS technology for processing 40 fps 640 × 360-resolution stereo video. The tile-based method saved 85% external memory with a power dissipation of only 288mW at 250MHz. The overall system in 65nm CMOS technology could run at 250MHz with 582mW for HD (1280 × 720)-resolution stereo video on the driving mode. The proposed processor in 40nm CMOS technology could produce 30 fps FHD depth maps at 170MHz with a power dissipation of 836mW and attained a 7% error rate under a 512-disparity search range (7bit for -disparity search range and 2bit for fraction) [28]. Recently, Li Z. *et al.* proposed a block-based SGM, which partitioned the image into several overlapped 50 × 50 pixels, achieved a 95.4% memory reduction for an FHD resolution. Then, they designed a neighbor-guided SGM fabricated in 28nm CMOS technology for dense stereo depth. It attained a processing speed of 30 fps at 180MHz and power dissipation of 760mW for 1920 × 1080-resolution stereo video under a 176-disparity search range [29].

III. The Proposed SGM With Region-Optimized Cost Aggregation Algorithm

A. Overall Description

In this work, we propose a regional-optimized SGM (rSGM) to solve the computational problem of cost aggregation.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

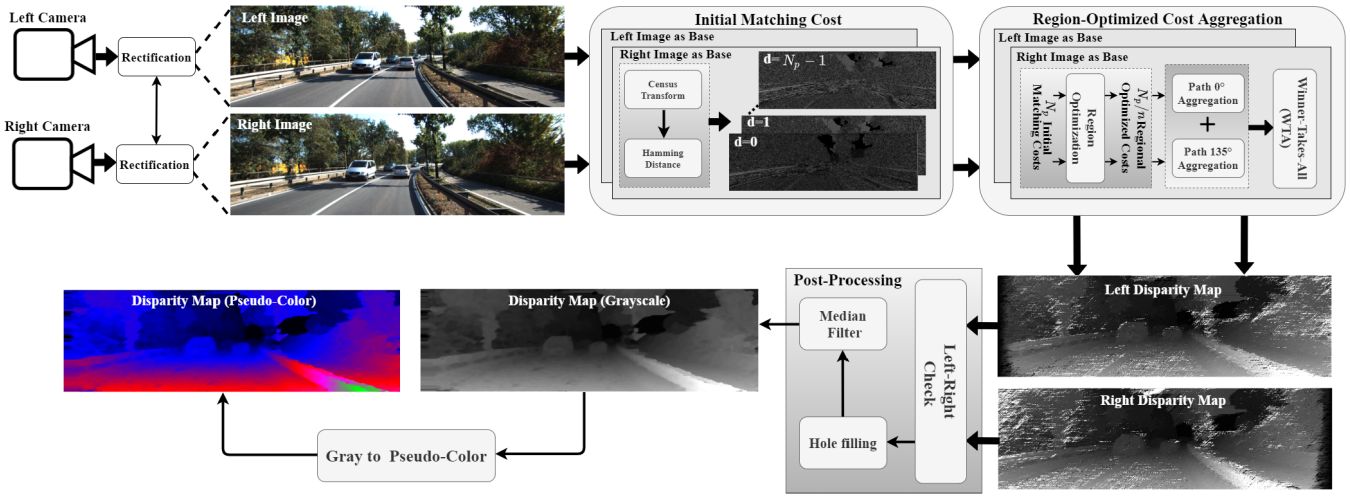


Fig. 1. Overall framework of the proposed rSGM with region-optimized cost aggregation.

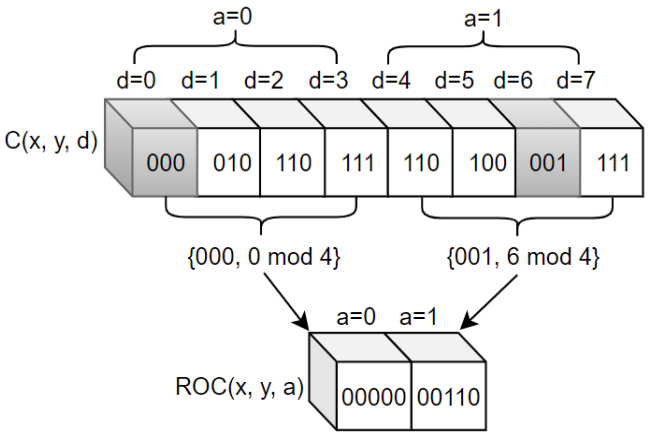


Fig. 2. Example of regional optimization with $N_p = 8, n = 4$ for a pixel $P(x, y)$.

As shown in Fig. 1, the proposed rSGM consists of four main modules: a rectification module, an initial matching cost computation module, a region-optimized cost aggregation module, and a post-processing module. The rectification, known as image transformation, is to obtain two rectified stereo images. The Initial matching cost calculation provides cost volume for the entire image under the disparity-search range. Then, the region-optimized cost aggregation optimizes the initial matching costs in a predefined region and aggregates in a pre-defined number of paths. Finally, the disparity is chosen by the WTA strategy, and several post-processing techniques are leveraged to further refine the disparity map and obtain the depth.

B. Rectification

In the stereo matching algorithm, the disparity of the pixel pair between the base and matching images is computed by searching the most similar pair horizontally. Therefore, the image pair must be rectified to set the corresponding pixels in the same line. However, a binocular camera system always has non-ideal image pairs caused by the camera distortion and baseline deviation.

To obtain a pair of rectified images, we need to rotate and translate images. In this work, an image transformation is applied to ensure the pair is on the same spatial plane and the heights are the same.

The image transformation is expressed by (1), where $(x/z, y/z)$ is the coordinates of a pixel in the rectified images and (u, v) is the coordinate of the input image.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = H_r \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{1}$$

here H_r is established based on the pre-measured parameters by the calibration process of two cameras. In the calibration with a MATLAB toolbox, multiple pairs of pixels on a calibration checkboard are selected. These pixels pairs are captured in different spatial positions to obtain the rotation matrix R and translation matrix T, which determine the spatial relationship of the two cameras' position and orientation in a specific three-dimensional space. Then the H_r can be obtained by $R \times T$.

C. Initial Matching Cost With Census Transform Algorithm

The Census Transform algorithm in [30] relies on the relative order of the gray value of the local area, which overcomes the error caused by the camera brightness deviation, improves the system robustness [31], and has low computational complexity. The main idea of the Census Transform algorithm is using a string of bits, which is called the Census transformation code, to represent the pixel window. The mapping relationship is expressed by (2), with $\otimes$ denoting a concatenation, P is a certain pixel in the image, W is the matching window centered on $P(x, y)$, which is generally a 3×3 or 5×5 square window.

$$R(P(x, y)) = \bigotimes_{(i,j)\epsilon W} \xi(P(x, y), P(x + i, y + j))$$
$$\xi(p, p') = \begin{cases} 1, & p < p' \\ 0, & p \geq p'. \end{cases} \tag{2}$$

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

Generally, a similarity between two pixels of the base and matching image is calculated and those with the highest correlation are assigned as corresponding. In this work, the Hamming distance is used to calculate the similarity after obtaining the Census transform codes of the pixels in the left and right images while the right image is supposed as the base. The Hamming distance is also called the initial matching cost and the smaller distance represents the greater the similarity.

The local matching cost $C\,(P\,(x,\,y),\,d)$ is calculated as (3), where x and y represent the horizontal axis and vertical axis respectively. $P(x, y)$ is a pixel located at $(x,\,y)$ in the base image, e.g., the right image is defined as the base image and $P_{(x+d,y)}$ is the candidate pixel in the left image, R_L and R_R are the respective Census transform codes in the left and right images. For d, whose range is $[0,\,N_p - 1]$, is a certain distance value between the base pixel and the candidate pixels in the matching image. Finally, the computation of the initial matching costs is conducted along Np candidate pixels.

$$C\,(x,\,y,\,\mathrm{d}) = \sum H(\mathrm{R}_R\left(P_{(x,y)}\right),\,R_L(P_{(x+d,y)})). \tag{3}$$

D. Regional Optimization and Cost Aggregation

Since the traditional SGM algorithm [32] aggregates the initial matching cost in different directions, the disparity search range and the cost aggregation paths are two main factors to the computational cost.

In this work, regional optimization is a strategy that treats a pre-defined number of pixels as a region. And then, the cost of this region for aggregation is represented by the minimum initial matching cost inside. The stereo disparity is estimated horizontally for every pixel in the base image within a disparity search range Np in the matching image. Thus, each pixel must repeat $Np \times Path$ times for aggregation where $Path$ is the number of aggregation paths. This certainly leads not only to high computational resources but also to huge memory for the subsequent calculation. In the case of surfaces with no texture or repetitive colors such as white walls and dark shadow, the stereo-depth estimations may yield a lot of redundant calculations of the cost aggregation.

The proposed rSGM is to decrease the number of the initial matching costs by a minimum search strategy among a region as in (4):

$$\begin{aligned} ROC\,(x,\,y,\,a) = \{&\min\left(C\,(x,\,y,\,(a,0)),\ldots,\right.\\ &\left.C\,(x,\,y,\,(a,n-1))\right),min_p\}, \end{aligned} \tag{4}$$

where a is the index of the cost region within $[0,\,\frac{Np}{n} - 1]$, $ROC(x, y, a)$ represents the region-optimized cost (ROC) of the region a, $C\,(x,\,y,\,(a,0)),\ldots,C\,(x,\,y,\,(a,n-1))$ represents n initial matching costs in the region a, min_p is the position of the minimal initial matching cost. In particular, min_p is associated with the least significant $\lceil log_2\ n\rceil$ bits of the minimum matching cost as the regional cost.

Taking $N_p = 8$ and $n = 4$ for a pixel $P\,(x,\,y)$ as an example in Fig. 2, each value in the cube stands for the initial matching costs of $P\,(x,\,y)$ with the disparity search range from 0 to 7. Subsequently, every 4 consecutive initial matching costs of $P\,(x,\,y)$ are considered as a region. The dark gray cube is denoting the minimum of them. Next, the position flag of each region is concatenated to the least significant 2 ($\lceil log_2\ 4\rceil$) bits of the cost. Finally, 8 initial matching costs are represented by 2 region-optimized costs. Therefore, the large amount of computation of the aggregation can be significantly reduced through optimizing Np initial matching costs of each pixel to $\frac{Np}{n}$ region-optimized costs. Thus, the larger n may lead to a higher error rate but lower resource consumption. Here, n can be scalable according to the accuracy requirement of an application.

The traditional SGM algorithm combines the global matching algorithm with the dynamic programming approach and simplifies the evaluation of the global energy function with the fixed directions' cost aggregation. In this work, the initial matching costs have been optimized to the regional costs through regional optimization. Consequently, for the direction r, the path cost is computed as in (5), where $L_r(p, a)$ represents the path costs for pixel point p in region a in direction r, $C\,(p, a)$ describes the regional cost for pixel p, and two penalties P_1 and P_2 are separately responsible for disparity changes and disparity discontinuities (P_1 is always less than P_2). The last term avoids steadily increasing path costs.

$$\begin{aligned} L_r\,(p,a) = {} & C\,(p,a) + \min[L_r\,(p-1,a)\,,L_r\,(p-1,a+1)\\ & + P_1,\min_{i\in[0,N_p-1]} L_r\,(p-1,i) + P_2]\\ & - \min_{i\in[0,N_p-1]} L_r(p-1,i). \end{aligned} \tag{5}$$

The final optimized costs are the sum of all the aggregation costs in different paths. The original SGM utilizes the two-scan method [28], which contains forward and backward scans, to perform eight-path cost aggregation. The paths in directions 0°, 45°, 90°, and 135° belong to the forward scan while the paths in 180°, 225°, 270°, and 315° directions belong to the backward scan. Generally, the optimal choice of aggregation paths differs in various SGM algorithms. Through our error rate analysis, the best choice of the paths of the proposed rSGM is the combination of 0° and 135° which can further solve the computational problem of the cost aggregation and save the hardware resource.

After the aggregation, the region with the minimum optimized cost is determined by the WTA strategy. Then, the real disparity $disp\,(P(x, y))$ of a pixel $P\ (x,\ y)$ can be restored from (6), where n is the number of costs in a certain region, R is the region of the minimum optimized cost and R_p is the corresponding position flag.

$$disp\,(P(x,\,y)) = R \times n + R_p. \tag{6}$$

E. Post-Processing

Post-Processing is crucial for obtaining dense and high-accuracy depth maps. We utilize left-right check, hole filling, and median filter to further refine the parallax. The left-right check approach is up to detect the occluded and badly matched pixel pairs through (7), where $D\,(P\,(x,\,y))$ means the validity of pixel $P\,(x,\,y)$ located at $(x,\,y)$, d_R is the value of the

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

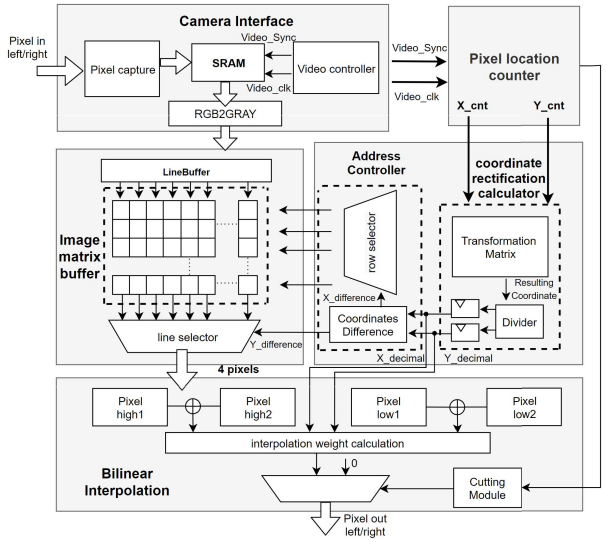


Fig. 3. Images rectification structure with resource-saving and pixel-level pipeline structure.

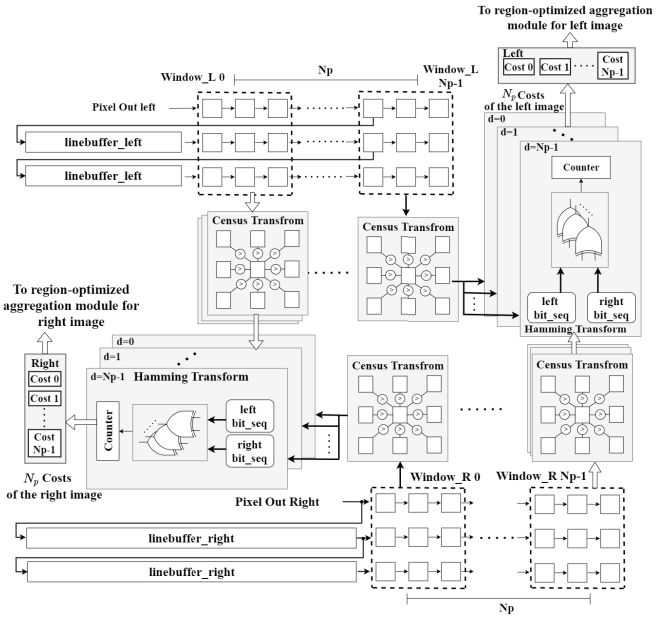


Fig. 4. Fully-parallel Census transform architecture to compute the Census codes with a pixel-level pipeline. The transform window size is 3×3 and the disparity searching rang is N_p. The post-processing of the left-right check is also implemented with fully-parallel architecture.

right disparity map at (x, y) while d_L is the value of the left disparity map at $(x + d_R, y)$,

$$D\left(P\left(x, y\right)\right) = \begin{cases} valid, & if \ \ |d_L - d_R| < 1 \\ invalid, & otherwise. \end{cases} \tag{7}$$

In the hole filling, the smaller value of the two nearest horizontal valid pixels of the invalid pixel is chosen to be the disparity. Then, an 11×11 median filter is utilized to obtain the final dense disparity map. Finally, the depth of a pixel $P(x, y)$ can be calculated according to (8), where $depth(P(x, y))$ is the depth of pixel $P(x, y)$, f is the pre-calibrated focal length, B is the baseline, c is the size of a pixel, and Z is the distance (depth) of the point from the camera.

$$depth(P(x, y)) = \frac{fB}{cZ}. \tag{8}$$

IV. Pixel-Level Pipeline Architecture for Region-Optimized SGM

A. Image Rectification With Pixel-Level Pipeline Structure

As described above, the rectification module is performed based on the transform matrix obtained by camera calibration. A resource-saving and pixel-level pipeline architecture is proposed as shown in Fig. 3. The coordinate rectification calculator takes the original pixel coordinates through the pixel counter, which counts the pixel location by the input *sync* signal, and calculates the rectified mapping coordinate, as described in (1). The resulting coordinate can be larger or smaller than the original coordinate due to the characteristics of rectified mapping. Therefore, multiple dual-port SRAMs are grouped to form an image-matrix buffer in which every pixel has an adjustment range, where the range is the number of pixels in the vertical direction and the horizontal direction.

However, the resulting coordinates are always not an integer. The *x_decimal* and *y_decimal* represent the decimal part of the resulting coordinates in the vertical and horizontal directions. According to the decimal part of the resulting coordinate, the *address controller* calculates the corresponding four addresses to read four interpolation pixels from the *image matrix buffer* to the *bilinear interpolation (BI)* module. After receiving the pixels and decimal part of coordinates, the *BI* module calculates the interpolation pixel and output the rectified image pixel streams.

B. Fully-Parallel Census Transform With Pixel-Level Pipeline

Each pixel in base and matching image, i.e., the right and left image in this work, needs to search in another image within the disparity range N_p. In this work, a fully parallel architecture for Census Transform is developed to compute the initial matching costs of pixels in the base and matching images which are shown in Fig. 4. The full parallelism means each pixel in the right image has N_p arithmetic units (*Window_L 0* to *Window_L* $N_p - 1$) for parallel computing the initial matching costs of the pixel in the left image.

In addition, N_p arithmetic units (Window_R 0 to Window_R $N_p - 1$) are implemented for the post-processing of the left-right check since the base image exchanges between the right and left image.

At first, two *linebuffer_rights* with the width of the image width W and two *linebuffer_lefts* with $W - N_p + 1$ width together with $N_p - 1$ windows are utilized to buffer the data of two rows of the base and matching images. Then, the computed $N_p \times 2$ Census codes are the left and right *bit_seq* in the *hamming transform* module. Next, $N_p \times 2$ Hamming distances are computed parallelly through XOR operation on each left and right *bit_seq*.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

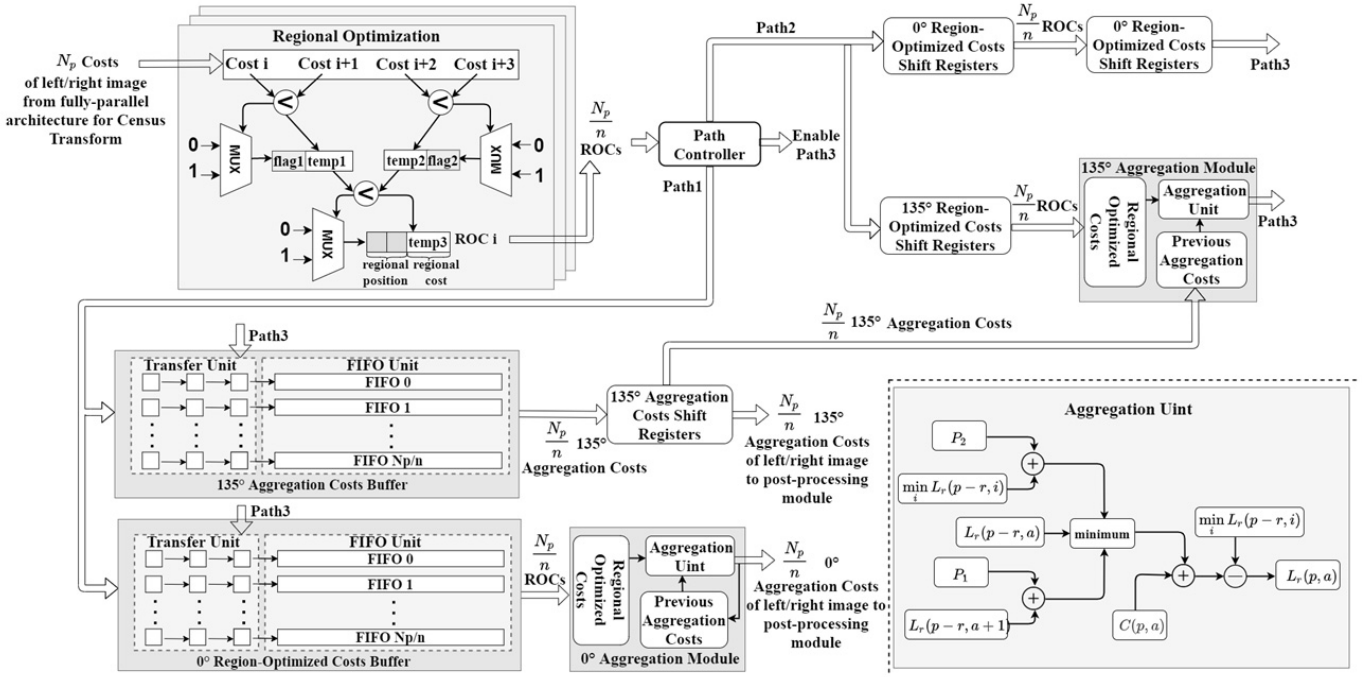


Fig. 5. Two-layer parallel two-stage pipeline (TPTP) structure to process the aggregation in two directions (According the error analysis, 0° and 135° can produce better depth result than four or eight directions). Here, the disparity searching range is N_p.

C. Regional Optimization and Cost Aggregation With Two-Layer Parallel Two-Stage Pipeline Structure

Each pixel yielding N_p initial matching costs lead to high computational complexity and large memory space for the cost aggregation. In this work, the regional optimization is leveraged to optimize the initial matching costs to the regional costs to reduce the resource consumption and improve the compactness of the hardware architecture. The regional optimization module in Fig. 5 firstly compresses N_p initial matching costs to N_p/n, as the example in Fig. 5 where the n is 4 ($Cost_i, \ldots, Cost_{i+3} (i = 0, 4, 8, \ldots, N_p - 4)$).

As described in Section II part D, the cost aggregation of the traditional SGM has forward and backward scans. The backward scan is opposite to the direction of the pixel stream. Thus, it is inevitable to cache the aggregated results of the forward scan in a frame buffer and then transfer them with the remaining four paths' results when the backward scan is accomplished. Therefore, the memory requirement is (*image size*) × (*disparity search range*) × (*bit width of the aggregated cost*) × (*number of forwarding paths*) × 2 where 2 is used for the left-right check for occlusion handling. Finally, 188MB is needed for 640 × 480 resolution, 128-disparity search range and 563MB is required for 1280 × 720 resolution.

To solve memory issue in eight-path aggregation, an 8 × 8 tile-based method is proposed in [26] which only buffers the first of every eighth row and then reconstruct the remaining data of the seven rows through forwarding aggregation in each tile. This method reduced the external bandwidth by 62.3% with the sacrifice of a 43.8% increment in on-chip computation. Besides, a block-based SGM algorithm partitioned the image into several overlapping 50 × 50 blocks which achieves 95.4% memory reduction for storing the forward scan aggregation and suffers only 0.5% accuracy degradation [28].

Traditionally, aggregation is utilized to minimize the 2-dimensional energy function through multiple 1-dimensional dynamic programming methods. From the experimental analysis, we find that the path of 135° is equivalent to the combination of 180° and 90°. As well as, 45° is equivalent to the combination of 0° and 90°. According to the error analysis, it is observed that the best aggregation path for rSGM is the combination of (0°, 135°) which is even better than four paths and eight paths aggregation.

In this work, we propose a two-layer parallel two-stage pipeline (TPTP) structure to process the aggregation in only two directions (0°, 135°) as shown in Fig. 5. Since the pixel follows the forward raster scan manner, the aggregation path for 135° has a row-level data dependence but the aggregation path for 0° has pixel-level data dependence. Thus, the FIFOs for 135° aggregation are used to buffer the intermediate aggregation of the last row while the FIFOs for 0° aggregation store the *ROCs* to match the timing of these two paths.

In detail, the *0° region-optimized costs buffer* is comprised of $3 \times N_p/n$ 7-bit registers. For the 7-bit, the lowest two bits record the relative position of the ROC and the highest five bits record the value of the ROC. And N_p/n FIFOs have W-3 words, each of which is 7-bit. Because the position flag of each region has already been recorded in the 0° submodule, the bit precision for the 135° submodule is only 5 bit. In this way, the memory usage to buffer the aggregated cost is decreased to (*image width*) × ($\frac{disparity\ range}{n}$) × (2 × *bit width of ROC* + *bit width of position flag*) × 2. Finally, only 0.0586MB and 0.117MB are respectively required for

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

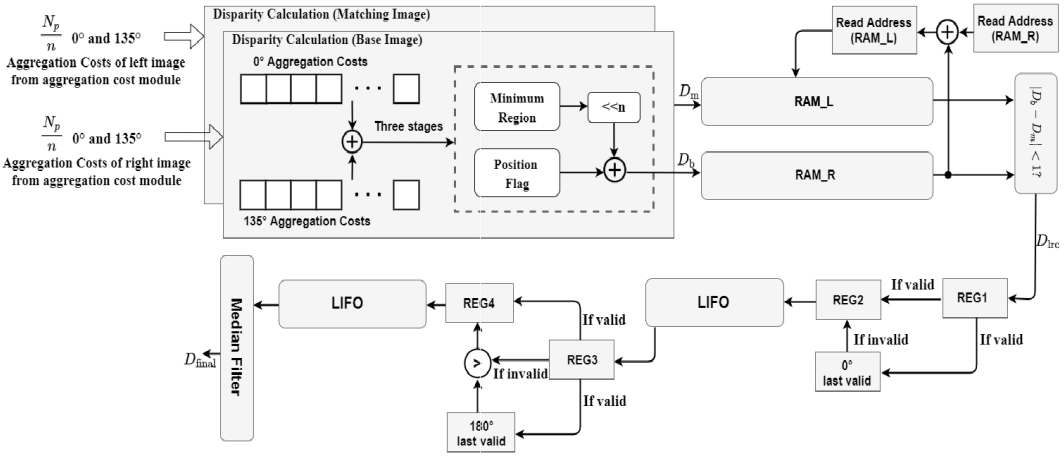


Fig. 6. Hardware implementation of post-processing based on pixel-level pipeline structure.

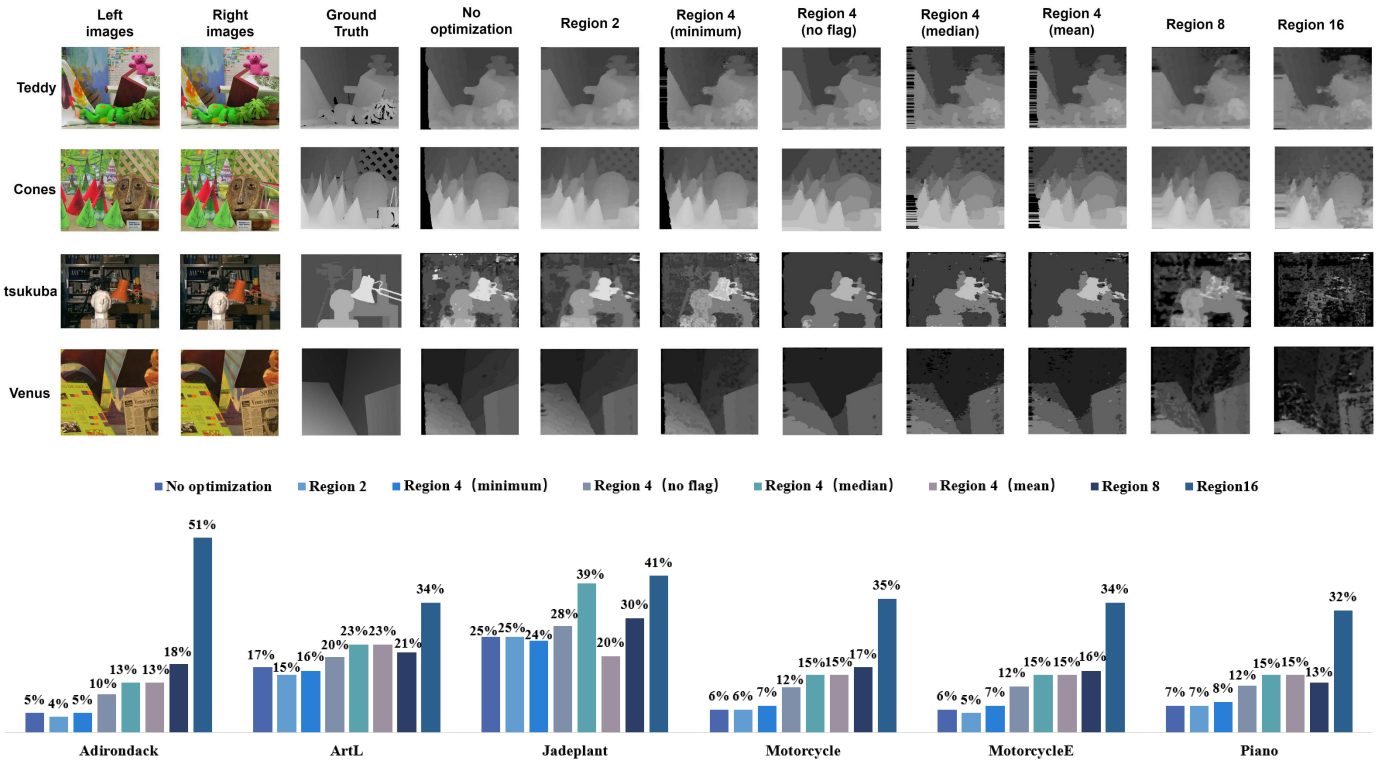


Fig. 7. Disparity images and percentage of bad pixels on the Middlebury V3 benchmark under $\sigma = 3$ in different-size region optimization.

resolution of 640×480 and 1280×720 under the 128-disparity search range, $n = 4$ and 7-bit width for *ROCs*. The rSGM achieves around 99.97% and 99.98% memory reduction.

Since the aggregation path for 135° has a row-level data dependence but the aggregation path for 0° has a pixel-level data dependence, the 0° aggregation path needs the *ROCs* of the current pixel and the aggregation costs of the previous pixel in the horizontal direction, the aggregation results of current pixel are returned to the *previous aggregation costs* block of the *0° aggregation module* for aggregation of next pixel which solves the dependencies in 0° direction. Here, the position flag in the 0° aggregation module does not participate in the calculation of the cost aggregation but is buffered temporally for later disparity calculation.

As for the path of 135°, the data dependence occurs when the cost aggregation of the current pixel requires the previous intermediate aggregation cost on the last row and the *ROCs* on the same row. It can be well solved by conveying the intermediate aggregation result in the 135° *aggregation costs shift registers* to the *previous aggregation costs* unit while the *ROCs* of the 135° *region-optimized costs shift registers* are concurrently passed to the *regional optimized costs* unit.

D. Depth Estimation and Post-Processing

The depth map associated with the base image (the right image) is determined by the disparity of a pixel and its matched pixel with the minimum cost. As well, the depth map

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

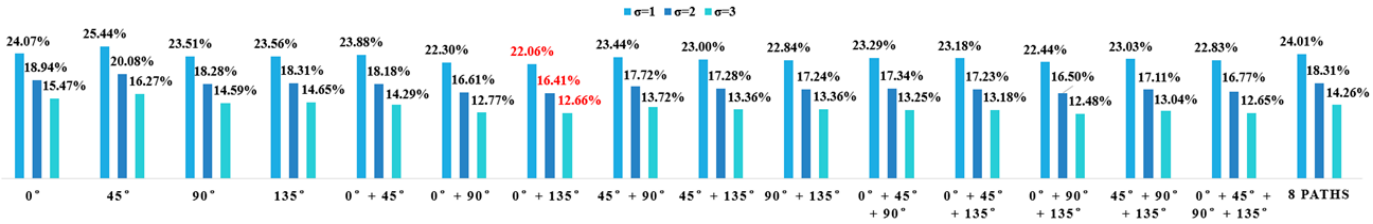


Fig. 8. Error rates of rSGM (Region size n is 4) in different combinations of aggregation directions **without** post-processing based on the Middlebury 3.0 benchmark under $\sigma = 1$, 2 and 3.

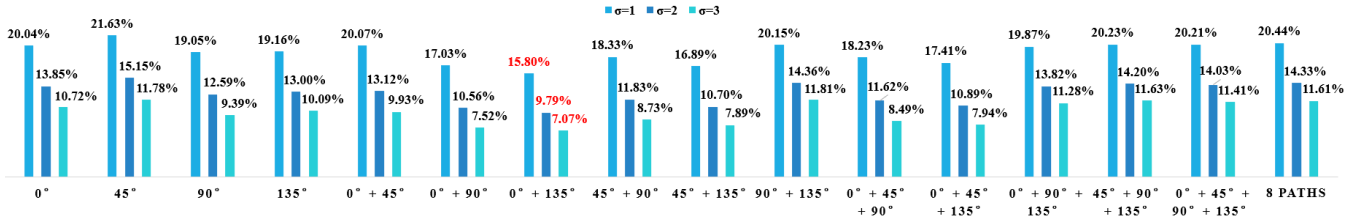


Fig. 9. Error rates of rSGM (Region size n is 4) in different combinations of aggregation directions **with** post-processing based on the Middlebury 3.0 benchmarck under $\sigma = 1$, 2 and 3.

TABLE I

COMPARISON OF THE ERROR RATE BETWEEN THE SGM AND RSGM WITH DIFFERENT NUMBER OF PATHS ON THE MIDDLEBURY 3.0 AND KITTI 2015 DATASETS

Algorithm	Middlebury V3			KITTI 2015	
	σ=1	σ=2	σ=3	noc	occ
SGM (0°+135°)	11.28%	8.48%	6.81%	5.60%	6.11%
SGM (4 paths)	15.06%	12.45%	10.96%	6.88%	8.21%
SGM (8 paths)	15.26%	12.45%	10.87%	6.93%	8.29%
rSGM (0°+135°)	15.80%	9.79%	7.07%	6.58%	7.27%
rSGM (4 paths)	20.21%	14.03%	11.41%	8.31%	9.65%
rSGM (8 paths)	20.44%	14.33%	11.61%	8.35%	9.67%

noc: non-occluded regions occ: all image regions

that corresponds to the matching image (the left image) can be determined from the same costs associated with the pixel q of the base image (the right image) with the minimum cost. The base and matching depth maps can handle the occlusions according to the false matches by left-right check. The final disparity is set to the minimum one if the disparity of a pixel in the base image is different from its corresponding disparity of this pixel in the matching image.

As shown in Fig. 6, the disparity is calculated by firstly summing up the 0° and 135° aggregation costs to attain the global energy in range $[0, \frac{N_p}{n} - 1]$. Besides, a three-stage pipeline is implemented to find the minimum value, i.e., the most matching region. Then left-shift by n is to recover the disparity back to the range $[0, N_p - 1]$.

Subsequently, two memory blocks with more than N_p words are used for the left-right check. Here, *Db* is the disparity of the base image and *Dm* is the disparity of the matching image. The *Dm* is found by adding *Db* and *Db*'s addresses together. Finally, the left-right check is implemented by comparing *Dm* and *Db* according to (7).

At last, the hole filling is utilized to replace the invalid pixel with a suitable value by finding the minimum of two nearest valid disparities in the 0° and 180° directions. We use two *LIFOs* (last in first out) to find these two valid disparities. In the scan for 0° direction, the value of the register *REG1* will be set to register *0° last valid* to record the most recent disparity value. Then shift the value of *REG1* to *REG2*. When the disparity in *REG1* is invalid, the value of *REG2* is directly set by register *0° last valid* while keep the mark invalid unchanged. Then, a *LIFO* is used to reverse the order of the initial pixel stream for the computation of the 180° direction. Finally, an 11×11 median filter using architecture in [33] is employed to smooth the final disparity map.

V. EXPERIMENTAL RESULT

A. Accuracy Analysis

The accuracy of the depth estimation is evaluated on the Middlebury V3 benchmark [34] which has 30 high-resolution image pairs with challenging conditions, such as slight rectification errors, different exposures, or different illuminations between the left and the right images. The evaluation metric in benchmark [10] is given by (9), where E is the accuracy of a certain image, W is image width, H is image height, N is $W \times H$, $d_r(i, j)$ is the disparity obtained by the proposed algorithm, $d_{gt}(i, j)$ is the ground truth disparity, and the σ is a threshold error value to determine the precision. This metric is also called the percentage of bad pixels whose error is greater than σ.

$$E = \frac{1}{N} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \left(\left| d_r(i, j) - d_{gt}(i, j) \right| > \sigma \right). \tag{9}$$

We also investigated the results of the no position flag, the median, and mean values in the region as shown in Fig. 7 with different region sizes and different optimization strategies.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

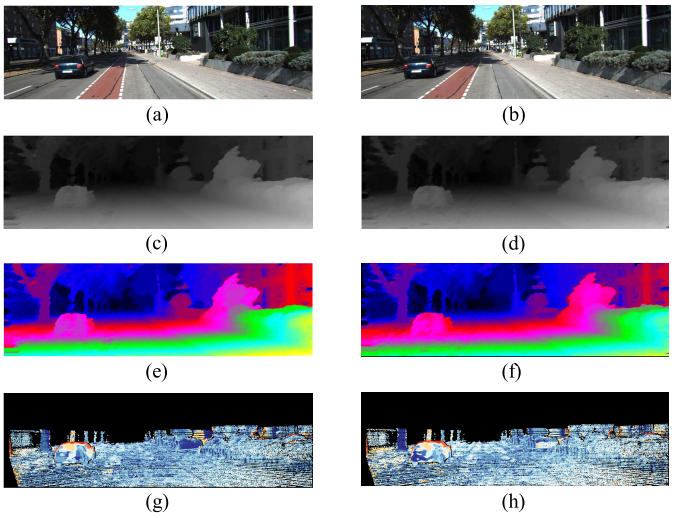


Fig. 10. (a and b) No.172 left and right images of KITTI2015 training sets; (c) Grayscale disparity map of traditional SGM; (d) Grayscale disparity map of rSGM with 4-size region and 0° + 135° paths; (e) Pseudo-color disparity map of traditional SGM; (f) Pseudo-color disparity map of rSGM with 4-size region and 0° + 135° paths; (g) Error map of traditional SGM (noc:5.58%, occ:6.60%); (h) Error map of rSGM(noc:6.49%, occ:7.52%); Disparity SNR of the traditional SGM(noc: 21.6 dB, occ: 20.8 dB); Disparity SNR of the rSGM(noc: 21.3 dB, occ: 20.6 dB).

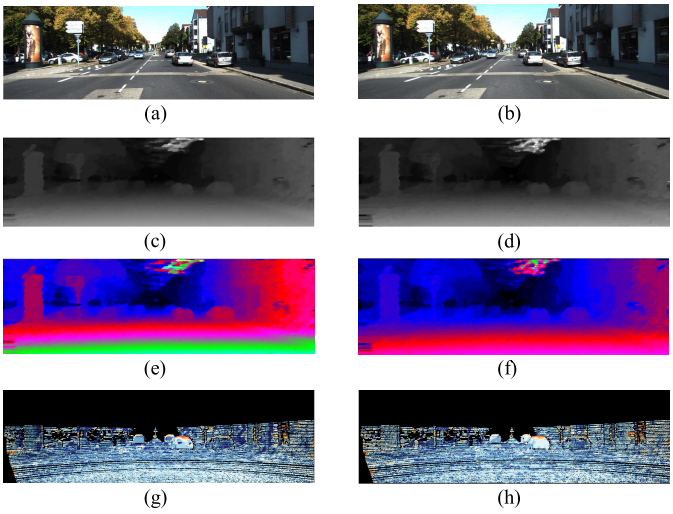


Fig. 11. (a and b) No.138 left and right images of KITTI2015 training sets; (c) Grayscale disparity map of traditional SGM; (d) Grayscale disparity map of rSGM with 4-size region and 0° + 135° paths; (e) Pseudo-color disparity map of traditional SGM; (f) Pseudo-color disparity map of rSGM with 4-size region and 0° + 135° paths; (g) Error map of traditional SGM (noc:3.27%, occ:3.83%); (h) Error map of rSGM(noc:4.14%, occ:4.88%). Disparity SNR of the traditional SGM(noc: 26.3 dB, occ: 25.6 dB); Disparity SNR of the rSGM(noc: 26.1 dB, occ: 24.2 dB).

From the visible results in the upper part of Fig. 7, we can find that the smoothy effect is apparent and the error rates are high in the case without the position flag. Finding the median value is implemented by sorting the n initial matching costs in the region and then taking the median one as the region-optimized cost of the corresponding region. The mean value of a region is calculated by summing up all the initial matching costs of a region and then dividing the result by n. It is observed that the quality of the disparity map becomes low while increasing the region size, and the optimization strategy of the minimum cost in each region attains the best disparity-map quality and accuracy in comparison to the median and mean strategy. As illustrated in Fig. 7, the regional optimization with the size of 4 leads to an average accuracy loss of less than 1% comparing to no optimization. In particular, the 2-size region-optimization achieves even better accuracy than the original SGM. As regional optimization is utilized to decrease resource consumption and accelerate the speed for satisfying the IoT edge devices, this loss is a trade-off between accuracy, resource, and speed.

Since different numbers and combinations of aggregation directions have disparate effects on the accuracy of the rSGM algorithm, we carry out the analysis of the error rates for the cost aggregation with different combinations of the directions. The error results on the latest Middlebury V3 benchmark under three threshold values $\sigma = 1$, 2, and 3 for investigating different combinations of the 4 aggregation directions (0°, 45°, 90°, and 135°) and 8 directions are shown in Fig. 8 and Fig. 9 which respectively present the results with and without the post-processing for the rSGM. The analysis results show that the error rates in eight directions are even worse than those in four directions. In particular, the error rates of the combination of 0° and 135° with post-processing,

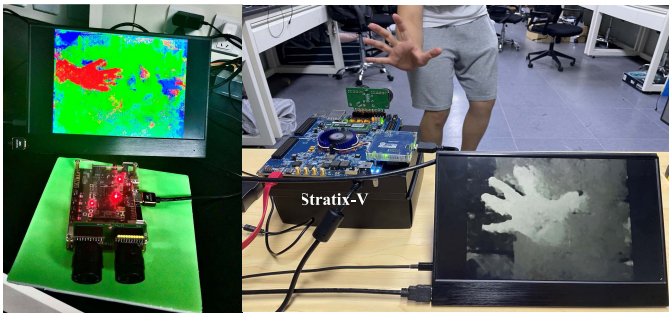


Fig. 12. Hardware implementation demonstrated on Xilinx Spartan-7 and Intel Stratix-V FPGA platforms.

i.e., 15.80%, 9.79%, and 7.07% under $\sigma = 1$, 2, and 3, are the best among these options.

Furthermore, we also test and compare the error rates of the traditional SGM and the rSGM with different numbers of aggregation paths on datasets (KITTI 2015 [35] and Middlebury 3.0) in Table I. From the results on Middlebury 3.0, it is observed that the SGM and rSGM with four-path and eight-path reach almost the same error rates under the same post-processing including left-right check, hole filling, and 11×11 median filter. In particular, the combination with only paths of 0° and 135° achieves the best accuracy. The reason is that the paths of 0° and 135° can reduce the accumulated error of the path decomposing.

In addition, we also evaluate the error rates of the SGM and rSGM on KITTI2015 training datasets under different scenarios where we select the same images as [33]. It is further proved that the paths of 0° and 135° still produce the best accuracy rate for both the SGM and rSGM. On this dataset, the result of the SGM is a little bit better than that of the

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISON WITH DIFFERENT ARCHITECTURE ON RESOURCE UTILIZATION AND PERFORMANCE

	Configuration			Resource Utilization			Performance			
Work	Resolution	Disparity Search Range	Platform	Area/ LUTs*	Gates/ Register *	Memory (bits)	Frequency (MHz)	Frame rate (fps)	Energy per pixel (nJ)#	Power (W)
ISSCC 2015 [25]	1920×1080	64	ASIC (40nm)	22 mm^2	1.50M gates	2,883,584	215	30	9.82	0.611
JSSC 2017 [26]	1280×720	64	ASIC (65nm)	16 mm^2	5.42M gates	4,041,605	250	30	11.9	0.330
SOCC 2017 [27]	640×480	64	ASIC (65nm)	6.3 mm^2	-	1,310,720	250	40	23.4	0.288
JSSC 2018 [28]	1920×1080	128	ASIC (40nm)	10.8 mm^2	-	1,089,536	170	30	13.4	0.836
JSSC 2019 [29]	1920×1080	176	ASIC (28nm)	9.3 mm^2	-	1,605,632	180	25	14.6	0.76
TCSII 2017 [21]	1024×768	64	Virtex-7	29,057 LUTs	-	2,726,298	50	63.57	3.44	0.172
JRTIP 2020 [17]	1024×768	128	Stratix-IV	141,000 LUTs	107,192 registers	6,291,456	125	25	198	3.9
TCSVT 2019 [24]	1920×1080	128	Kintex-7	53,190 LUTs	40,980 registers	5,556,464	148.5	60	-	-
TCSVT 2021 [33]	1280×960	128	VCU-118	54,573 LUTs	48,191 registers	4,626,432	200	116	-	-
ASSCC 2018 [23]	1980×1088	128	ZYNQ7000	167,206 LUTs	67,000 registers	8,865,792	54	30	5.94	0.384
This work	1024×768	128	Stratix-IV	31,851 LUTs	16,675 registers	806,933	109	139	5.1	0.556
							24	30	4.96	0.119
	1024×768	128	Virtex-7	29,955 LUTs	30,459 registers	1,087,488	127	162	3.76	0.478
							24	31	2.42	0.058
	1280×960	128	VCU-118	33,206 LUTs	31,140 registers	700,416	209.7	170.7	2.28	0.478
							24	25	2.05	0.063
	1920×1080	128	Kintex-7	52,200 LUTs	36,707 registers	1,087,488	127	61	6.11	0.776
							24	12	3.92	0.094
	1980×1088	128	ZYNQ7000	52,200 LUTs	36,707 registers	1,087,488	127	59	6.08	0.772
							24	11	3.75	0.090
	640×480	128	Stratix-V	17,863 LUTs	18,197 registers	507,737	156	508	4.29	0.669
							24	78	4.38	0.105
	1024×768	128	Stratix-V	18,000 LUTs	18,171 registers	808,940	153	195	4.31	0.660
							24	31	4.29	0.103
	1920×1080	128	Stratix-V	18,175 LUTs	18,001 registers	1,514,496	154	74	4.39	0.676
							24	12	4.38	0.105
	640×480	128	Spartan-7	29,797 LUTs	31,327 registers	622,592	109	355	4.74	0.517
							24	78	2.63	0.063

*The area and gates are the metrics of ASIC platform while the LUTs and Register are the metrics of FPGA platform
#: Energy/pixel =Power/(frame per second × Resolution)

rSGM. The results indicate that the accuracy of the rSGM can satisfy the requirement of robot navigation and autonomous cars.

According to the above analysis, we finally choose rSGM with a 4-size region and the combination of the 0° and 135° for aggregation. Although the rSGM with a 2-size region can achieve high accuracy that is even better than traditional SGM, the rSGM with a 4-size region can save hardware resources and memory usage.

Furthermore, as for the noise performance, the comparison between the disparity map and error map of two KITTI 2015 training datasets (No.172 and No.138 with noise), where the red and yellow parts mean badly matched pixel for the default 3-pixel threshold, is illustrated in Fig. 10 and Fig. 11. These results indicate that most errors occur at the place where the gradient changes greatly and the interior of the object is almost always matched correctly. Since the regional optimization with a local minimum of a region still must be aggregated in a disparity searching range, it works as a smooth filter that blurs the gradient edges. Furthermore, the signal-to-noise ratio (SNR) is expressed to quantify the noise performance in (10) where H and W are the image height and width, $d_r\,(i,\,j)$ is the disparity obtained by the proposed algorithm, and $d_{gt}\,(i,\,j)$ is the ground truth disparity. The SNR comparison results of No.172 and No.138 training images in KITTI 2015 are shown in Fig. 10 and Fig. 11, the SNR of SGM is slightly higher than the rSGM. Therefore, the proposed rSGM can well inherit the robustness and accuracy of the traditional SGM and has great improvement on the computational complexity and hardware-resource consumption.

$$SNR = 10\log_{10}\left[\frac{\sum_{i=1}^{H}\sum_{i=1}^{W} d_{gt}\,(i,\,j)^2}{\sum_{i=1}^{H}\sum_{i=1}^{W}\left[d_{gt}\,(i,\,j) - d_r\,(i,\,j)\right]^2}\right]. \quad (10)$$

B. Discussion of the Hardware Implementation

As described above, a pixel-level pipeline architecture is designed for the rSGM. Accordingly, the processing speed can be deduced according to (6), where *fps* represents frame

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

TABLE III
ERROR RATE COMPARISON TO THE STATE-OF-ART WORKS ON KITTI 2015 AND MIDDLEBURY V3 DATASETS

Work	KITTI 2015	Middlebury V3		
		σ=1	σ=2	σ=3
JSSC 2018[28]	7%	-	-	-
JSSC 2019[29]	7.52%	-	-	-
JRTIP 2020[17]	-	32.9%	-	-
TCSVT 2019[24]	6.88%	16.19%	-	-
TCSVT 2021[33]	7.44%	-	-	-
This work	7.27%	15.80%	9.79%	7.07%

per second, f_{max} is the maximum frequency and *res* is the image resolution.

$$fps = \frac{f_{max}}{res}. \tag{11}$$

In this work, the depth estimation coprocessor with the proposed rSGM is demonstrated on a low-cost Xilinx Spartan-7 FPGA and an advanced Intel Stratix-V FPGA platform in Fig. 13 with MT9V034 global-shutter CMOS Image Sensors (CISs). In Fig. 12, the disparity maps in pseudo-color and grayscale maps express the distances where the color from red to blue or from light to dark indicates the distance from near to far. For the case of VGA resolution with a 128-disparity searching range and a size of 4 for regional optimization, it consumes only 29K LUTs, 31K Registers, and 622Kbit on-chip memory on the Spartan-7 FPGA and 17K LUTs, 18K Registers, and 496Kbit on-chip memory on the Stratix-V FPGA. In particular, the max working frequency can reach up to 156 MHz, which means a processing speed of 508 fps, and the power dissipation is 669mW. The dynamic power dissipation is only 63 mW at a typical working frequency of 24MHz for about 60-fps real-time processing.

Table II presents the comparison results of the hardware implementation between the proposed architecture and various state-of-art hardware implementations for depth estimation with the SGM stereo matching. This work is synthesized on different FPGA families to eliminate the effect on the FPGA type where an advanced FPGA can produce a better performance. It is observed that the proposed coprocessor for depth estimation significantly outperforms the previous FPGA-based state-of-art implementations on logic-resource consumption (LUTs), memory usage, and processing speed. With the same constraints such as the 128-disparity searching range, the image resolution, and the FPGA platform, the memory usage of thixs work is 1/5 as that in [24] and 1/7 as that in [17]. Since the resolution of our coprocessor only affects the size of the memory for implementing the line buffer, our work still occupies less hardware resource than that in [21] even with a smaller disparity of 64.

As in [17], except for their hardware usage with almost 8× memory, 6× registers, and 4× LUTs, the working frequency with 125MHz is much higher than that in this work with 110MHz under the Stratix-IV platform but their speed with 25 fps is more than 5× slower than that in this work. Thus, the pixel-level pipeline architecture significantly solves the calculation bottleneck of the cost aggregation and accelerates the depth estimation.

The power dissipation is estimated by XILINX Vivado and Intel Quartus Prime tools according to the FPGA type, hardware-resource usage, and clock frequency. Due to the pixel-level pipeline architecture synchronizing to the working frequency of the image sensor, except for the capacitance related to the scale of the hardware resource (chip area), the dynamic power is mainly affected by the working frequency. Consequently, this work is the most energy-efficient among the FPGA implementations under the same conditions. It is also comparable to the ASIC implementation in 28 nm CMOS technology with 6.72nJ/pixel (VGA) and 14.6nJ/pixel (FHD) [29] which can achieve 30 fps for stereo depth at VGA and 25 frames/s at FHD resolution. Although several ASIC designs in Table II improved the SGM through different optimization to save the memory utilization, they adopted the external memory. For large bandwidth, these works need high working frequency. Therefore, the memory usage in [28] is almost the same to this work, their processing speed of 30fps with the working frequency of 170MHz is at least 2× slower than a speed of 61 fps with the working frequency of 127 MHz.

In [16], a complicated CNN is used to find the optimal penalty coefficients on NVIDIA Titan X GPU with the power of about 250W and they gain a high-quality disparity maps output. In addition to the GPU implementation, combining an advanced FPGA with CPU in [17] is also not suitable for IoT applications.

The accuracy, which is often expressed by the error rate, is compared to the state-of-art works on the two mainstream datasets KITTI 2015 and Middlebury V3. The results in Table III show that the error rate of this work is comparable to that of the others on the KITTI 2015 datasets, and it is even a little better than that in [17] and [24] on Middlebury V3 datasets. This indicates that this work with compact hardware architecture also ensures accuracy.

VI. CONCLUSION

In this paper, we have proposed a region-optimized SGM algorithm and designed a compact stereo-matching hardware architecture. Regionalized initial matching costs have significantly avoided large resource consumption in the aggregation process. The two-layer parallel two-stage pipeline structure in this work allows the calculation of two directions (0° and 135°) aggregation to solve the crucial computational bottleneck of the SGM algorithm and this significantly saves resources even without external memories. In the IoT area, stereo-matching devices need to be compact, economical, and fast. Comparing with the previous works, our hardware implementation used the least resources with limited accuracy loss demonstrated on the low-cost FPGA device XILINX Spartan-7 and a high-level Stratix-V FPGA device. It is observed that our architecture is the most energy-efficient than the state-of-art works that make it directly comparable to the ASIC implementation.

REFERENCES

[1] K. Lu, X. Wang, Z. Wang, and L. Wang, "Binocular stereo vision based on OpenCV," in *Proc. IET Int. Conf. Smart Sustain. City*, Shanghai, China, 2011, pp. 1–4, doi: 10.1049/CP.2011.0312.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.

[2] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 993–1008, Aug. 2003, doi: 10.1109/TPAMI.2003.1217603.

[3] D. Shi and Y. Li, "Depth extraction method based on binocular stereo matching," in *Proc. 4th Int. Congr. Image Signal Process.*, Shanghai, China, 2011, pp. 1420–1423, doi: 10.1109/CISP.2011.6100436.

[4] K. Takaya, "Stereo disparity measurement for binocular stereo video systems," in *Proc. ICCAS-SICE*, Fukuoka, Japan, 2009, pp. 2648–2651.

[5] J.-K. Oh, S. Lee, and C.-H. Lee, "Stereo vision based automation for a bin-picking solution," *Int. J. Control, Autom. Syst.*, vol. 10, no. 2, pp. 362–373, Apr. 2012.

[6] Y. Zhao, X. Hou, L. Jia, and S. Ma, "The obstacle avoidance system for mobile robot based on binocular stereo vision," in *Proc. 8th World Congr. Intell. Control Autom.*, Jinan, China, Jul. 2010, pp. 6461–6465, doi: 10.1109/WCICA.2010.5554283.

[7] A. Seki and M. Okutomi, "Robust obstacle detection in general road environment based on road extraction and pose estimation," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, 2006, pp. 437–444, doi: 10.1109/IVS.2006.1689668.

[8] C. G. Keller, M. Enzweiler, M. Rohrbach, D. F. Llorca, C. Schnorr, and D. M. Gavrila, "The benefits of dense stereo for pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 1096–1106, Dec. 2011, doi: 10.1109/TITS.2011.2143410.

[9] M. Li, L. K. Kwoh, C.-J. Yang, and S. C. Liew, "3D building extraction with semi-global matching from stereo pair worldview-2 satellite imageries," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Milan, Italy, Jul. 2015, pp. 3006–3009, doi: 10.1109/IGARSS.2015.7326448.

[10] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proc. IEEE Workshop Stereo Multi-Baseline Vis. (SMBV)*, Kauai, HI, USA, Apr. 2001, pp. 131–140, doi: 10.1109/SMBV.2001.988771.

[11] S. Sarika, V. A. Deepambika, and M. A. Rahman, "Census filtering based stereomatching under varying radiometric conditions," *Procedia Comput. Sci.*, vol. 58, pp. 315–320, Jan. 2015.

[12] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1582–1599, Jul. 2009, doi: 10.1109/TPAMI.2008.221.

[13] B. Chen and H.-P. Chen, "A realization of mutual information calculation on GPU for semi-global stereo matching," in *Proc. 5th Int. Conf. Intell. Netw. Intell. Syst.*, Tianjin, China, Nov. 2012, pp. 113–116, doi: 10.1109/ICINIS.2012.14.

[14] J. Ma, W. Yin, C. Zuo, S. Feng, and Q. Chen, "Real-time binocular stereo vision system based on FPGA," in *Proc. 6th Int. Conf. Opt. Photon. Eng. (icOPEN)*, Jul. 2018, Art. no. 108271U.

[15] F. Bethmann and T. Luhmann, "Object-based multi-image semi-global matching-concept and first results," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XL-5, pp. 93–100, Jun. 2014.

[16] A. Seki and M. Pollefeys, "SGM-nets: Semi-global matching with neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6640–6649, doi: 10.1109/CVPR.2017.703.

[17] L. F. S. Cambuim, L. A. Oliveira, E. N. S. Barros, and A. P. A. Ferreira, "An FPGA-based real-time occlusion robust stereo vision system using semi-global matching," *J. Real-Time Image Process.*, vol. 17, no. 5, pp. 1447–1468, Oct. 2020.

[18] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation," in *Proc. Int. Conf. Embedded Comput. Syst., Archit., Modeling Simulation*, Samos, Greece, Jul. 2010, pp. 93–101, doi: 10.1109/ICSAMOS.2010.5642077.

[19] S. Jin *et al.*, "FPGA design and implementation of a real-time stereo vision system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 15–26, Jan. 2010, doi: 10.1109/TCSVT.2009.2026831.

[20] C. Ttofis, C. Kyrkou, and T. Theocharides, "A low-cost real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2678–2693, Sep. 2016, doi: 10.1109/TC.2015.2506567.

[21] L. Puglia, M. Vigliar, and G. Raiconi, "Real-time low-power FPGA architecture for stereo vision," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 11, pp. 1307–1311, Jul. 2017, doi: 10.1109/TCSII.2017.2691675.

[22] L. F. S. Cambuim, J. P. F. Barbosa, and E. N. S. Barros, "Hardware module for low-resource and real-time stereo vision engine using semi-global matching approach," in *Proc. 30th Symp. Integr. Circuits Syst. Design Chip Sands (SBCCI)*, Fortaleza, Brazil, 2017, pp. 53–58.

[23] L.-D. Chen, Y.-T. Lu, Y.-L. Hiao, B.-H. Yang, W.-C. Chen, and C.-T. Huang, "A 95pJ/label wide-range depth-estimation processor for full-HD light-field applications on FPGA," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2018, pp. 261–262, doi: 10.1109/ASSCC.2018.8579289.

[24] X. Zhang, H. Sun, S. Chen, L. Song, and N. Zheng, "NIPM-sWMF: Toward efficient FPGA design for high-definition large-disparity stereo matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1530–1543, May 2019, doi: 10.1109/TCSVT.2018.2833743.

[25] H.-H. Chen, C.-T. Huang, S.-S. Wu, C.-L. Hung, T.-C. Ma, and L.-G. Chen, "23.2 A 1920×1080 30fps 611 mW five-view depth-estimation processor for light-field applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2015, pp. 1–3, doi: 10.1109/ISSCC.2015.7063106.

[26] K. J. Lee *et al.*, "A 502-GOPS and 0.984-mW dual-mode intelligent ADAS SoC with real-time semiglobal matching and intention prediction for smart automotive black box system," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 139–150, Jan. 2017, doi: 10.1109/JSSC.2016.2617317.

[27] K. Bong, K. Lee, and H.-J. Yoo, "A 590MDE/s semi-global matching processor with lossless data compression," in *Proc. 30th IEEE Int. Syst. Chip Conf. (SOCC)*, Sep. 2017, pp. 18–22, doi: 10.1109/SOCC.2017.8225998.

[28] Z. Li *et al.*, "A 1920×1080 30-frames/s 2.3 TOPS/W stereo-depth processor for energy-efficient autonomous navigation of micro aerial vehicles," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 76–90, Jan. 2018, doi: 10.1109/JSSC.2017.2751501.

[29] Z. Li, J. Wang, D. Sylvester, D. Blaauw, and H. S. Kim, "A 1920×1080 25-frames/s 2.4-TOPS/W low-power 6-D vision processor for unified optical flow and stereo depth with semi-global matching," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1048–1058, Apr. 2019, doi: 10.1109/JSSC.2018.2885559.

[30] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision* (Lecture Notes in Computer Science), vol. 801, J. O. Eklundh, Ed. Berlin, Germany: Springer, 1994, pp. 151–158.

[31] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 467–474, doi: 10.1109/ICCVW.2011.6130280.

[32] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.

[33] Z. Lu, J. Wang, Z. Li, S. Chen, and F. Wu, "A resource-efficient pipelined architecture for real-time semi-global stereo matching," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Feb. 26, 2021, doi: 10.1109/TCSVT.2021.3061704.

[34] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 8753, X. Jiang, J. Hornegger, and R. Koch, Eds. Cham, Switzerland: Springer, 2014, pp. 31–42.

[35] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3061–3070.



Pingcheng Dong is currently pursuing the graduation degree with the School of Microelectronics, Southern University of Science and Technology, Shenzhen, China.

He is a member of the Intelligent Sensing Laboratory, School of Microelectronics, Southern University of Science and Technology. His current research interests include image processing, stereo vision, machine learning, and VLSI circuits design.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.



Zhuoyu Chen is currently pursuing the graduation degree with the School of Microelectronics, Southern University of Science and Technology, Shenzhen, China.

He is a member of the Intelligent Sensing Laboratory, School of Microelectronics, Southern University of Science and Technology. His current research interests include image processing, digital integrated circuit design, stereo vision, and machine learning.



Zhuoao Li is currently pursuing the graduation degree with the School of Microelectronics, Southern University of Science and Technology, Shenzhen, China.

He is a member of the Intelligent Sensing Laboratory, School of Microelectronics, Southern University of Science and Technology. His research interests include digital integrated circuit design, computer vision, and machine learning.



Yuzhe Fu is currently pursuing the graduation degree with the School of Microelectronics, Southern University of Science and Technology, Shenzhen, China.

He is a member of the Intelligent Sensing Laboratory, School of Microelectronics, Southern University of Science and Technology. His current research interests include image processing, digital integrated circuit design, and deep learning.



Lei Chen received the B.S. and M.S. degrees from Qingdao University of Science and Technology in 2006 and 2009, respectively, and the Ph.D. degree from Hiroshima University, Higashihiroshima, Japan, in 2012.

In October 2012, she pursued her post-doctoral research at the HiSIM Research Center, Hiroshima University. From 2018 to 2020, she worked with Sharp Company Ltd. She then worked with the Pengcheng Laboratory, Shenzhen, China, from 2020 to 2021. She is currently a Research Associate Professor at the Southern University of Science and Technology, Shenzhen. Her research interests include circuit design and hardware algorithm design for image processing and video compression.



Fengwei An (Member, IEEE) received the B.S. degree from Qingdao University of Science and Technology in 2006, the M.S. and Ph.D. degrees from Hiroshima University, Japan, in March 2010 and March 2013, respectively.

Since 2013, he worked with Hiroshima University. From 2018 to 2019, he worked with Panasonic Semiconductor. He is currently an Associate Professor at the Southern University of Science and Technology. His research interests include energy-efficient image recognition algorithms and low-power circuits for embedded systems.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on November 17,2026 at 14:07:49 UTC from IEEE Xplore. Restrictions apply.