

Implementation of Real-Time Post-Processing for High-Quality Stereo Vision

Seungmin Choi, Jae-Chan Jeong, Jiho Chang, Hochul Shin,
Eul-Gyoon Lim, Jae Il Cho, and Daehwan Hwang

We propose a novel post-processing algorithm and its very-large-scale integration architecture that simultaneously uses the passive and active stereo vision information to improve the reliability of the three-dimensional disparity in a hybrid stereo vision system. The proposed architecture consists of four steps — left-right consistency checking, semi-2D hole filling, a tiny adaptive variance checking, and a 2D weighted median filter. The experimental results show that the error rate of the proposed algorithm (5.77%) is less than that of a raw disparity (10.12%) for a real-world camera image having a $1,280 \times 720$ resolution and maximum disparity of 256. Moreover, for the famous Middlebury stereo image sets, the proposed algorithm's error rate (8.30%) is also less than that of the raw disparity (13.7%). The proposed architecture is implemented on a single commercial field-programmable gate array using only 13.01% of slice resources, which achieves a rate of 60 fps for $1,280 \times 720$ stereo images with a disparity range of 256.

Keywords: Stereo vision, 3D depth, FPGA, post processing, hole filling, variance check, weighted median filter.

I. Introduction

Many studies on stereo vision for obtaining three-dimensional information have been conducted [1]–[4]. Since the release of Microsoft Kinect (2009), in particular, researchers are paying more attention to applications that use 3D depth sensors. However, applications using 3D depth information have trouble detecting long and thin objects, such as a human finger, at a distance of more than 3 m owing to the performance margins of 3D depth sensors. To overcome this limitation, Jeong and others proposed a stereo matching system using a time-division pattern projection [1], and Chang and others modified this system for implementation in four field-programmable gate arrays (FPGAs) for real-time processing [2]. Chang and others stated that their systems can calculate the depth for a $1,280 \times 720$ resolution image at 60 fps in an indoor environment [2]. The 3D system in [2] shows excellent depth results in a normal indoor home, but the authors claimed that it might suffer from noises, or outliers, in a disparity map when there are a number of occlusions or textureless regions in input images.

To counter a weakness of the previous studies, we propose a novel combination of post-processing algorithms, which is useful for refining raw disparity data. In addition, we propose a hardware-friendly post-processing architecture and its implementation result in an FPGA in the later sections. Firstly, in our proposal, the general left-right consistency checking is used to remove mismatched points. Then, we propose a real-time compact 2D hole filling (HF) method to fill the holes that have inaccurate depth values caused by mismatching at occlusion regions. In particular, we call this semi-2D HF, because we fill up the holes with adjacent background pixels in three directions. Furthermore, we propose a tiny variance

Manuscript received Dec. 11, 2014; revised 22 June, 2015; accepted 8 June, 2015.

This work was supported by the ICT R&D program of MSIP/IITP, Rep. of Korea (11921-03001, Development of beyond Smart TV Technology).

Seungmin Choi (corresponding author, ccsmm@etri.re.kr), Jae-Chan Jeong (channij80@etri.re.kr), Jiho Chang (changjh@etri.re.kr), Hochul Shin (creatrix@etri.re.kr), Eul-Gyoon Lim (eg_lim@etri.re.kr), Jae Il Cho (jicho@etri.re.kr), and Daehwan Hwang (hdh@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

checking logic using a mean deviation (MD) in substitution of the variance, which can reduce the number of multiplication operations needed to obtain a square value of a pixel. Moreover, this tiny variance checking logic uses the active pattern scene and the passive object scene at the same time for improving performance in searching textureless regions. Lastly, we propose a 2D weighted median filter (WMF) that uses the similarity and proximity weight of a passive scene and disparity of a hybrid matching to reduce the outlier noise occurring from mismatching. Because this is an edge-preserving noise reduction filter, the results may not suffer from smoothing artifacts. The proposed median filter has been designed using a novel pipelined three-stage cumulative histogram.

This paper is composed of six sections. Section II provides an introduction to the overall stereo system used to verify the proposed post-processing algorithms. Section III describes the novel post-processing algorithms. In Section IV, we show its experimental evaluations with data sets having ground-truth. In Sections V and VI, we describe the hardware architecture and its FPGA implementation, respectively. Finally, we provide some concluding remarks in Section VII.

II. System Overview

In this chapter, we describe the algorithms used in a stereo matching system implemented through a previous work [1]–[2]. In addition, we introduce the proposed post-processing algorithm in the next section.

Figure 1 shows a flow chart of the active stereo matching system including the proposed post-processing method at the

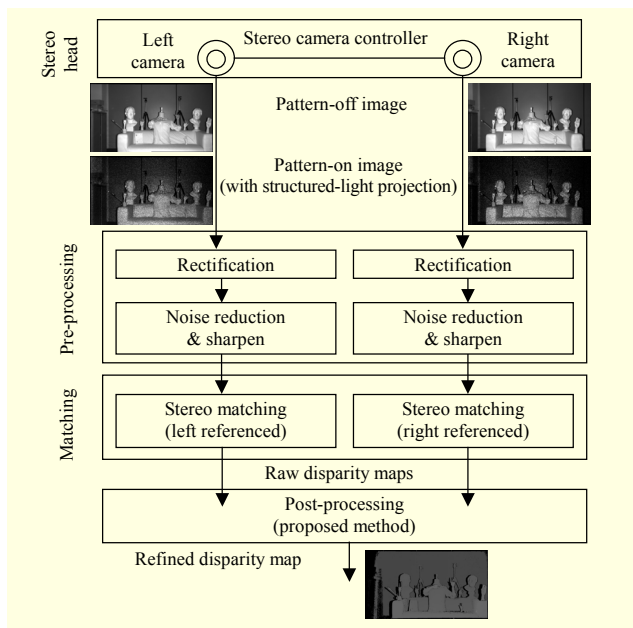


Fig. 1. Overall flow of stereo correspondence process.

last stage. The system consists of a “stereo head” module, pre-processing logic, stereo matching logic, and the post-processing logic proposed in this paper.

1. Stereo Head

We use two E2V661 CMOS sensors having a good quantum efficiency in the IR band to build the depicted stereo head. The resolution of the stereo image is $1,280 \times 720$ pixels (high-definition (HD) resolution) at a rate of 60 fps. Image streams from the stereo head are transferred to an FPGA board through an LVDS BUS. The stereo head also receives control signals from a computer through a USB 3.0 BUS. The synchronization control signals are transferred to the laser diode (LD) and light-emitting diode (LED) modules, which indicate the on/off time cycle of the LD and LED devices.

An active pattern is projected by a diffractive optical element, which was designed to diffract an 808 nm laser beam in the IR bandwidth. The projector uses a pseudorandom pattern designed by taking into account the brightness and density of the active pattern. It also has LEDs of the same IR band, and through it stereo cameras can acquire non-pattern images. The reason we use the IR band is because it is invisible to the human eye and allows us to control the light source.

2. Pre-processing

During the pre-processing stage, tasks such as image sharpening, noise removal, and rectification are conducted for image streams incoming from the left and right cameras. A bilateral filter, which is a non-linear, edge-preserving, and noise-reducing smoothing filter [3], is only applied to the “pattern-off image” used for guiding or aggregating the cost values. On the other hand, an un-sharp masking filter having an image sharpening effect is applied to both a “pattern-on image” and a “pattern-off image,” which are used for calculating the raw cost volume. The mask filter values in Table 1 are optimized to reduce the hardware resources for the divider by using a bit shift operation as a substitute for a divider in a convolution process. In addition, we use the Caltech method [5] to perform rectifications of the stereo images in the pre-processing stage.

Because the rectification task is one of the most important

Table 1. Pre-processing algorithms.

Function	Algorithms
Rectification	Caltech toolbox [5]
Noise reduction filter	Bilateral filter [3]
Sharpening filter	Mask: $-1 \ -2 \ -1; -2 \ 28 \ -2; -1 \ -2 \ -1$

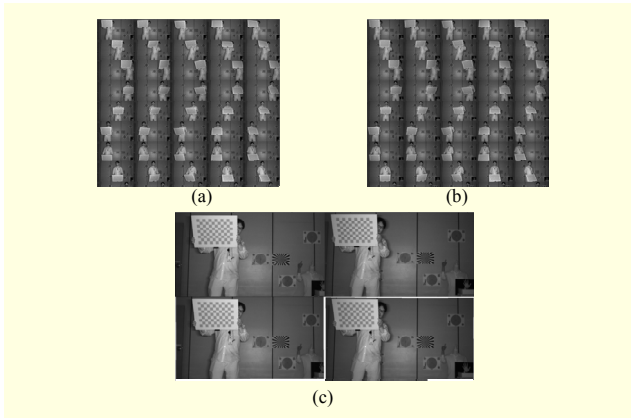


Fig. 2. Images used in rectification and their results in proposed system: (a) left camera (54 images), (b) right camera (54 images), and (c) example of left image/right image; its rectified left image/right image.

Table 2. Parameters used in rectification.

Parameter		Description
Intrinsic	fc	Focal length, 2×1 matrix
	cc	Principal point, 2×1 matrix
	alpha_c	Skew coefficient, scalar value
	kc	Distortion coefficient, 5×1 matrix
Extrinsic	Om	Rotation coefficient, 3×1 matrix
	Tc	Translation coefficient, 3×1 matrix

parts of a stereo matching process when considering the epipolar constraint, we use one hundred and eight images taken by the stereo head facing a checker board at various locations to extract the rectification parameters [1]. Figure 2 shows the images used in our rectification process. Camera Calibration Toolbox, developed by Caltech [5], has been widely used in many researches of stereo vision. It provides the best performance among the released software. Intrinsic parameters generated in the process of calibration are shown in Table 2 and are generated independently from the left and right cameras. Each calibration parameter — fc, cc, alpha, and kc — represents a camera’s internal elements, such as focal length, principal point, asymmetric coefficient, radial distortion, and tangential distortion. Extrinsic parameters, Om and Tc, are calculated using variables obtained from stereo cameras and can describe both the rotation and the translation transformation of the two coordinate systems, respectively. Rectification processes implemented here are described in [5]–[6] in detail.

3. Stereo Matching Algorithm

The general stereo matching algorithm consists of matching

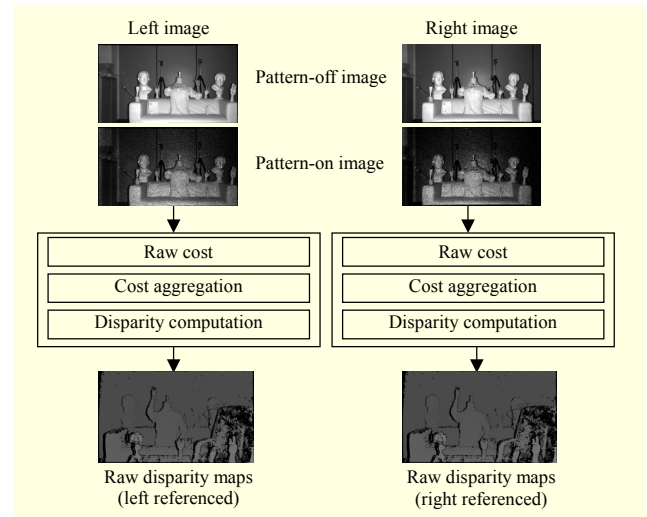


Fig. 3. Stereo matching algorithm.

cost computations (raw cost), cost aggregation, and disparity computations [4]. In addition to such processing, we also need to take passive and active images into account in calculating and aggregating the cost volume at the same time [1]. Figure 3 shows the overall active stereo matching algorithm used in the system. The stereo matching algorithm comprises the cost computations, cost aggregation, and disparity computations.

A. Cost Computation (Raw Cost, AD-CT)

We calculate the raw cost volume using “the absolute intensity difference (AD) – Hamming distance of census transform (CT),” the AD-CT, in this paper. The reason for combining the AD and the Hamming distance of CT cost measures is that the AD-CT provides better matching accuracy than either the individual AD measure or CT measure [7]. The AD-based cost function, $C_d^{AD}(p)$, and CT-based cost function, $C_d^{CT}(p)$, are calculated as follows:

$$C_d^{AD}(p) = \sum_{c \in \{Intensity\}} |I_c(p) - I_c(p-d)|, \quad (1)$$

$$C_d^{CT}(p) = \text{HammingDistance}(CT(p), CT(p-d)), \quad (2)$$

where d is disparity and $I_c(p)$ and $CT(p)$ are the intensity and CT value at a pixel p . In addition, the combined raw cost, $C_d(p)$, is obtained by

$$C_d(p) = \alpha C_d^{AD}(p) + (1-\alpha) C_d^{CT}(p), \quad (3)$$

where α balances the AD and CT terms.

B. Cost Aggregation

According to [8] and [9], the cost aggregation is a correction process for raising the discriminative of the raw cost. The aggregated cost, $C_d^{AG}(p)$, is expressed as a multiplication of

the support weight, $w(p, q)$ and raw cost, $C_d(q)$, as follows:

$$C_d^{AG}(p) = \sum_{q \in \omega_p} w(p, q) C_d(q), \quad (4)$$

where p is a pixel whose depth needs to be estimated, q is a neighboring pixel of p , and ω_p is the window centering on pixel p .

Supporting weight-based cost aggregation carries with it certain problems, such as computational complexity and a lengthy execution time. Thus, various methods that reduce the computational complexity and operate rapidly have been proposed in [10]–[11]. Herein, domain transform (DT) aggregation is used owing to its hardware-friendly property. For DT aggregation, dX_1 and dY_1 of an image are the horizontal and vertical gradients at the intensity domain, and σ_s and σ_r are the spatial (proximity) parameter and intensity range (similarity) parameter, respectively [11].

$$a = \exp\left(-\frac{1}{\sigma_s}\right), \quad (5)$$

$$g_x = 1 + \frac{\sigma_s}{\sigma_r} dX_1, \quad (6)$$

$$g_y = 1 + \frac{\sigma_s}{\sigma_r} dY_1. \quad (7)$$

The cost aggregation is performed in the horizontal direction (left-to-right and right-to-left). Using the results from the horizontal direction, then, the cost aggregation is performed in the vertical direction (top-to-bottom and bottom-to-top), as follows:

$$C_{d,y}^{AG-L}[x] = C_{d,y}[x] + a^{g_x} C_{d,y}^{AG-L}[x-1], \quad (8)$$

$$C_{d,y}^{AG-R}[x] = C_{d,y}^{AG-L}[x] + a^{g_{x+1}} C_{d,y}^{AG-R}[x+1], \quad (9)$$

$$C_{d,x}^{AG-T}[y] = C_{d,x}[y] + a^{g_y} C_{d,x}^{AG-T}[y-1], \quad (10)$$

$$C_{d,x}^{AG-B}[y] = C_{d,x}^{AG-T}[y] + a^{g_{y+1}} C_{d,x}^{AG-B}[y+1], \quad (11)$$

where $C_{d,y}^{AG-L}$, $C_{d,y}^{AG-R}$, $C_{d,x}^{AG-T}$, and $C_{d,x}^{AG-B}$ are the aggregation cost of left-to-right, right-to-left, top-to-bottom, and bottom-to-top, respectively. In addition, $C_{d,x}^{AG-B}$ in equation (11) is the final result of the cost aggregation.

C. Disparity Computation

Winner-takes-all, which is a local minimization method, is applied to find the minimum cost on a pixel-by-pixel basis. The estimated disparity, $d(p)$, is then found by

$$d(p) = \arg \min_{d \in S_d} (C_{d,x}^{AG-B}(p)), \quad (12)$$

where S_d is the set of all possible disparities and $C_{d,x}^{AG-B}(p)$ is the aggregated cost.

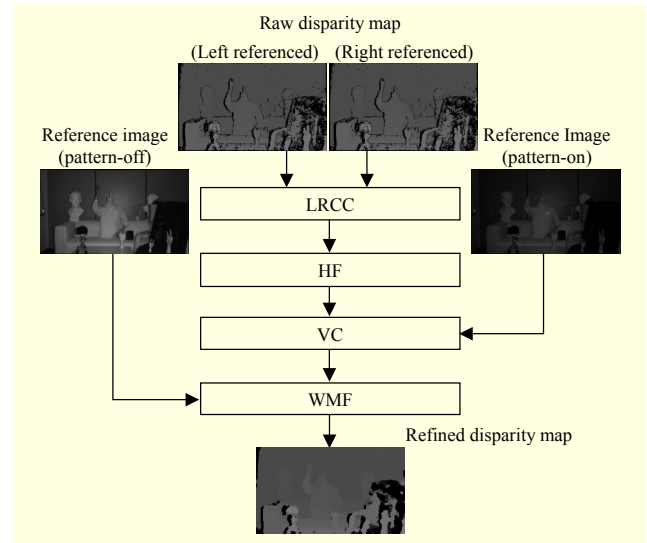


Fig. 4. Proposed post-processing algorithms.

III. Proposed Post-Processing Algorithms

Figure 4 shows a flow chart of the post-processing algorithms proposed in this paper. We previously introduced these algorithms and their results in [12]–[13], but more detailed explanations are described here. The post-processor's input consists of two raw disparity maps including the occlusion region from the left-referenced stereo matcher and the right-referenced stereo matcher. In the first place, the Left-Right Consistency Check (LRCC) makes the disparity map into a single channel by removing the occlusion region. In the second place, 2D HF fills the holes caused by the above occlusion checking process with the adjacent disparity inferred background. In addition, in the next step, we use a variance check (VC) to remove the output in the textureless region. In the last step, the WMF is used for removing the streak noise in the disparity map. Note that it is an advantage to locate the noise reduction filter after the VC because it can also fill a hole caused the VC.

1. LRCC

A consistency check is conducted with the left and right disparity maps. If the difference between the left and right disparity in the same index exceeds a predefined threshold then that disparity is an invalid value caused by an occlusion or a mismatching point.

$$DLRC(p) = |D_L(p) - D_R(p_d)|, \quad (13)$$

$$d_{lrcc}(p) = \begin{cases} D_L(p) & DLRC(p) < th_{lrcc} \\ -1 & DLRC(p) \geq th_{lrcc} \end{cases}, \quad (14)$$

where p is a pixel in the reference image, $DLRC(p)$ is the

disparity map after consistency checking, $D_L(p)$ is the left-referenced disparity at pixel p , and $D_R(p_d)$ is the right-referenced disparity at pixel p_d shifted from p with d or $D_L(p)$. If the DLRC(p) of the corresponding pixel is less than or equal to a threshold (th_{lrc}) fixed in advance, then the output, $d_{lrc}(p)$, should become $D_L(p)$, which means this disparity has a high confidence at this point. Otherwise, the output should become “-1,” which means its value is invalid; thus, it has to be defined as a hole.

2. 2D HF

The occlusion area in stereo vision is defined as the region observed from only one of two images. Because, in most cases, an occlusion area is located in the background region, it is simple and useful to choose the nearest background disparity for a hole where an occlusion has been removed [14]. There will be no artifacts in the output disparity map as long as the stereo scene remains in the front-to-parallel plane [15]. To choose the nearest background disparity for a hole, it is necessary to sample the adjacent valid disparities in eight directions to find the minimum disparity inferred background. However, because it requires a high complexity to implement in hardware, we use its approximation for searching in three directions to find the minimum value. After sampling the non-hole disparities in three directions, it is necessary to select the minimum disparity among them to fill in the hole.

3. VC

Having low uniqueness in the cost function in (11), the textureless areas are more likely to be generally mismatched. Because the low uniqueness tends to cause a monotonous energy graph, it is difficult to determine a disparity having the minimum value in the energy graph, and it may thus cause a noisy disparity map. Therefore, it is important to find the textureless region to treat this problem.

A VC is a way to use the variance as a criterion for making a decision on the existence of texture in a region. Thus, we can easily remove a disparity having a low confidence in a textureless region with the VC. We use the “pattern-on image” to calculate the variance and remove the region having a variance below a predefined threshold. Equation (15) is a function used to calculate the window-based variance value, and (16) is the function used to calculate the (absolute) MD in substitution of the variance for a simple hardware implementation.

$$\sigma^2(p) = \frac{1}{N} \sum_{i \in NP_p} (x_i - \mu_{NP_p})^2, \quad (15)$$

$$MD(p) = \frac{1}{N} \sum_{i \in NP_p} |x_i - \mu_{NP_p}|, \quad (16)$$

where N is the size of a window currently processed, NP_p means the neighborhood pixels in the window, x_i is a pixel's intensity, and μ_{NP_p} is the mean of the neighborhood pixels in the window.

4. WMF

A median filter is generally used to remove high-frequency noise such as “salt and pepper” noise. This is useful to remove a noise disparity and interpolate a hole with adjacent valid disparities [16]. Applying a median filter to a disparity map, we need to assume that there are sufficient valid disparities in the window including a noise disparity. However, if a window is located near the center of an object's edge, then the selected medial value may be incorrect. Thus, the disparity near the edge would suffer from a blurring effect with a general median filter.

To avoid this problem, we designed a novel edge-aware WMF having the coefficients used in a bilateral filter [3] by modifying the conventional WMF in [17]. For an effective hardware implementation, we use a median filter based on a cumulative histogram, as in [18]. To construct a histogram, the first step is to *bin* the range of values, and then count how many values fall into each interval. A rectangle is drawn with a height proportional to the count and width equal to the bin size in the histogram graph. Finally, we integrate the histogram to obtain the cumulative histogram. In addition, the median value is the index of the total count/2. The main difference between a general median filter and WMF is that one has to add not (+1) but (+weight) for the bin counter when the sample hits this bin.

We use the Gaussian function for the weight, as with a bilateral filter, because it considers the similarity and proximity for the image. Equation (17) is the weight for the median filter used in this paper.

$$w_{p,q} = \exp\left(-0.5 \times \left(\frac{\Delta c_{p,q}}{\sigma_s}\right)^2\right) \times \exp\left(-0.5 \times \left(\frac{\Delta g_{p,q}}{\sigma_d}\right)^2\right), \quad (17)$$

where $\Delta c_{p,q} = I(p) - I(q)$, $\Delta g_{p,q} = |p - q|$.

IV. Experimental Evaluation of Proposed Algorithm

In this chapter, we show the experimental results of a simulation with the proposed algorithm before implementing its hardware architecture. In the experiments, we use two types of dataset, one is the well-known Middlebury stereo image set [4], [19], and the other is a real-world stereo image set used in [1], to find the optimal parameter for the best performance and hardware implementation. The purpose of the first experiment is to discover the best performance of the proposed algorithm with the well-known passive stereo image sets, and the second

experiment is for finding an optimal parameter to be used in designing a very-large-scale integration (VLSI) with a real-world active stereo image set having ground-truth.

1. Experiments Using Passive Middlebury Stereo Image Sets

Table 3 shows a comparison of the proposed algorithm for four well-known Middlebury stereo image sets in [4], [19]. When our algorithm is used in the experiment, the error rate for all pixels of the four image sets is 8.3%, which is lower than Jeong's (13.7%) [1] and Jin's (17.2%) results [20].

2. Experiments Using an Active Real-World Stereo Image Set

Before designing the hardware architecture, it is necessary to find the optimal parameters, such as the window size, similarity,

proximity, and threshold (see Table 4). To find them, we evaluated the algorithms with an active real-world stereo image set having the ground-truth used in [3], where its full resolution is $1,280 \times 854$, as shown in Fig. 5(e), the valid resolution of the ground-truth is $1,020 \times 600$, as shown in Fig. 5(c), and its disparity range is 256. Using these optimal parameters after evaluation, we could achieve a high performance and cost-effective usage of hardware resources in designing the VLSI concurrently.

The percentage of bad pixels, or error rate (that is, pixels whose absolute disparity error is greater than one), is 10.16% for all pixels and 4.67% for non-occluded pixels. Because there is no post-processing task in occlusion regions to complement them in the ground-truth [1], we concentrated on the results of the percentage of bad pixels for not non-occluded pixels but all pixels in the following experiments. In addition, for the same reason mentioned above, we do not apply HF to a raw disparity map in the evaluation experiment using the ground-truth in [1]. After applying the LRCC to the raw disparity, the error rate for all pixels decreases from 10.16% to 6.26%, and its optimal threshold is equal to a value of three, as shown in Fig. 6. After applying the VC to the raw disparity from LRCC, the error rate for all pixels decreases from 6.26% to 6.12% when its optimal

Table 3. Results of Middlebury dataset.










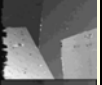
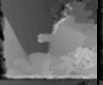



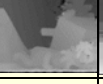
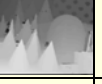
Image sets [4] and [19]	Tsukuba	Venus	Teddy	Cones		
Left image					Average ratio (%) of bad pixels	
Ground truth						
AD-CT in hybrid [1]						
Proposed (using only intensity)						
Bad pixels (% all)	Jin's [20]	11.56	5.27	21.50	17.58	17.24
	Jeong's [1]	11.67	6.62	19.60	16.73	13.7
	Proposed	6.55	1.66	14.9	10.8	8.3

Table 4. Optimal parameters for proposed logic.

Module	Parameter	Optimal	Error (%) (for all pixels, threshold=1)	Effect
Raw disparity	N.A	N.A	10.12	Before applying our algorithms
LRCC	Threshold	3	6.26	Removing occlusion region
VC	Window size	9×9	6.12	Removing texture-less region
	threshold	5.5		
WMF	Window size	7×7	5.77	Edge-preserving spark noise reduction
	Sigma of proximity	33		
	Sigma of similarity	3		
	Iteration	1		

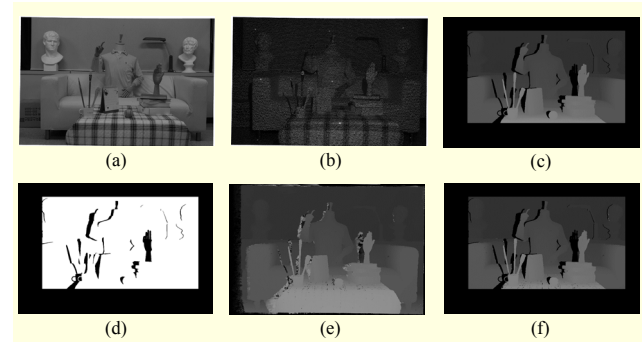


Fig. 5. Ground-truth of real-world image set: (a) pattern-off image (left), (b) pattern-on image (left), (c) ground truth, (d) occluded region map (black pixels are in occluded region), (e) raw disparity map, and (f) valid parity map (non-occlusion) [3].

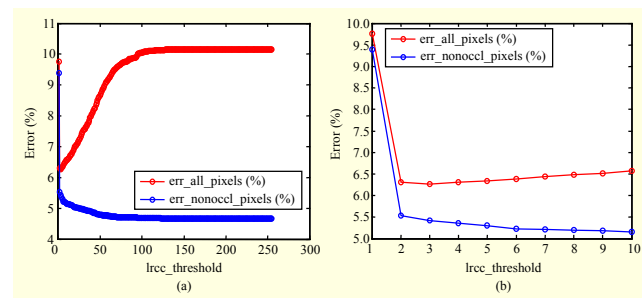


Fig. 6. Parameter for LRCC: (a) threshold from 1 to 255 and (b) threshold from 1 to 10.

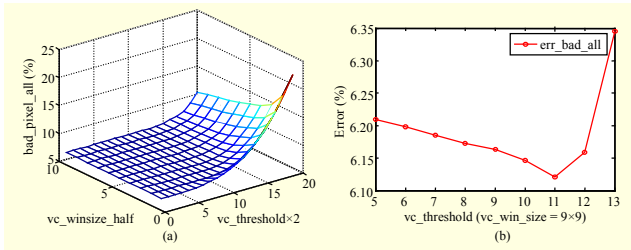


Fig. 7. Parameters for VC: (a) window size and threshold vs. error and (b) threshold vs. error (window size = 9×9 , threshold = 11×0.5).

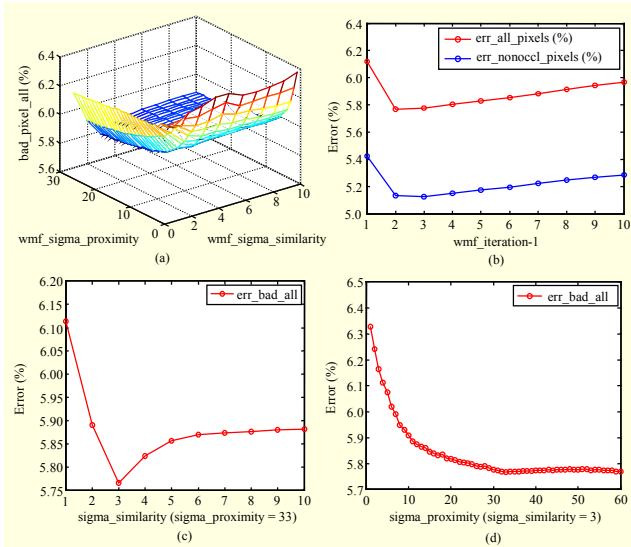


Fig. 8. Parameters for WMF: (a) sigma (proximity and similarity) vs. error, (b) iteration vs. error, (c) similarity vs. error (with fixed proximity), and (d) proximity vs. error (with fixed similarity).

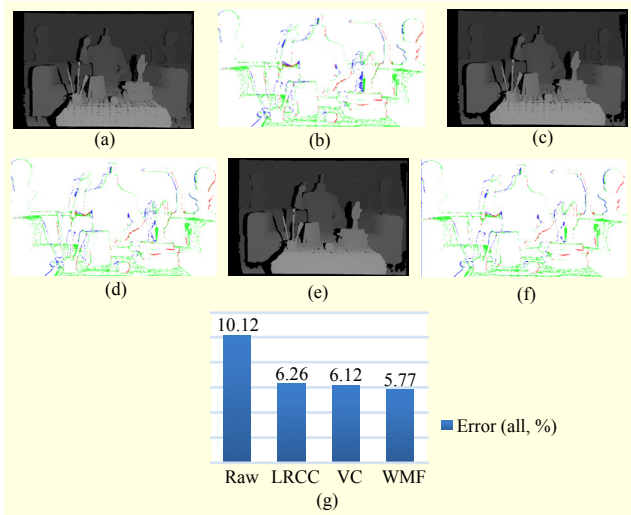


Fig. 9. Optimal parameters and their results: (a) LRCC (threshold = 3), (b) error of LRCC = 6.26, (c) VC (window 9×9 , threshold = 5.5), (d) error of VC = 6.12, (e) WMF (window 7×7 , sigma proximity = 33, sigma similarity = 3, iteration = 1), (f) error of WMF = 5.77, and (g) comparison of errors.

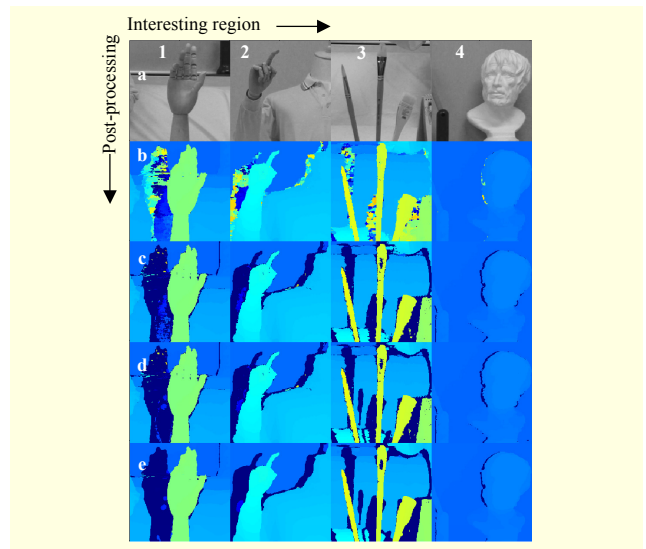


Fig. 10. Refinement for four regions having thin structure: (a) image, (b) raw disparity map, (c) LRCC, (d) VC, and (e) WMF.

parameters are a window size of 9×9 and a threshold of 5.5, as shown in Fig. 7. After applying the WMF to the raw disparity processed by VC, the error rate for all pixels decreases from 6.12% to 5.77%, where the optimal parameters for the filter are a window size of 7×7 , a sigma value for a proximity of 33, a sigma value for a similarity of 3, and an iteration count of 1, as shown in Fig. 8. The complete results are shown in Figs. 9 and 10 (resizing the region of interest), and the optimal parameters are reported in Table 4.

V. Hardware Design for Proposed Algorithms

We implemented the algorithms proposed in this paper into a single FPGA. In this chapter, we introduce the hardware architecture in detail. The proposed post-processing hardware consists of four sub-blocks in a cascaded manner, which are a consistency check, HF, a VC, and a WMF. Figure 11 shows the top architecture of the proposed post-processing hardware. It uses a normal intensity image with no structured-light patterns to calculate the weight coefficients for a WMF. On the other hand, it uses a structured-light image to calculate a local window's variance for a VC, because a region having no structured-light pattern might have low confidence in the disparity map, as in a textureless region in passive stereo vision.

Because the HF may have some artefacts near the hole, its function needs to be excluded for a real-world environment. In addition, unlike the Middlebury image sets captured in a limited number of environments, there are a large number of environments in the real world. Thus, the optimal parameter values for the proposed algorithms could vary depending on the scene. For the reason mentioned above, we added internal

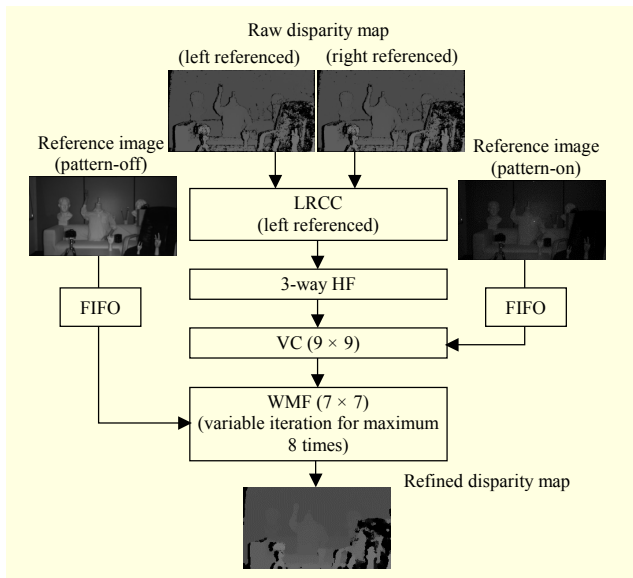


Fig. 11. Proposed hardware architecture for post-processing.

memory registers to control the optimal parameters of algorithms in the hardware design except the tap size of the filter. Table 3 shows the optimal parameters derived from the previous experiments for active real-world images to be used in designing a VLSI. When we apply the proposed algorithm to the raw disparity using the parameters in Table 3, the error rate decreases from 10.12% to 5.77% for the image set.

1. 3-Way HF

We designed 1-way HF logic, initially, and applied it to the vertical and horizontal directions to design a 3-way HF logic. Figure 12(b) shows the pseudocode of RTL for the 1-way HF logic, which finds the minimum nonzero disparity inferred as the nearest background disparity in a particular direction. In addition, Fig. 12(a) shows the top scheme of 3-way HF logic using the 1-way HF mentioned above. The 1-way HF scans and processes a line image from left to right. To reuse the 1-way HF for a right-to-left processing, extra logic is needed; that is, a horizontal mirror block that is able to convert the horizontal axis. After all, the 1-way HF for the left-to-right direction obtains input data, or disparity, directly, while that for the right-to-left direction needs a horizontal-mirror-block flipping axis of the input data in the horizontal direction before obtaining the input data to reuse the 1-way HF.

In the case of the top-to-bottom direction, the HF logic needs to buffer a line to reuse the 1-way HF. In addition, at the “Minimum” stage in Fig. 12(a), it needs to synchronize the three data paths, top-to-bottom, left-to-right, and right-to-left. Let $t(p)$ be the clock tick used in the processing of the 1-way HF, and let $t(L)$ be the clock used in buffering a line. In addition,

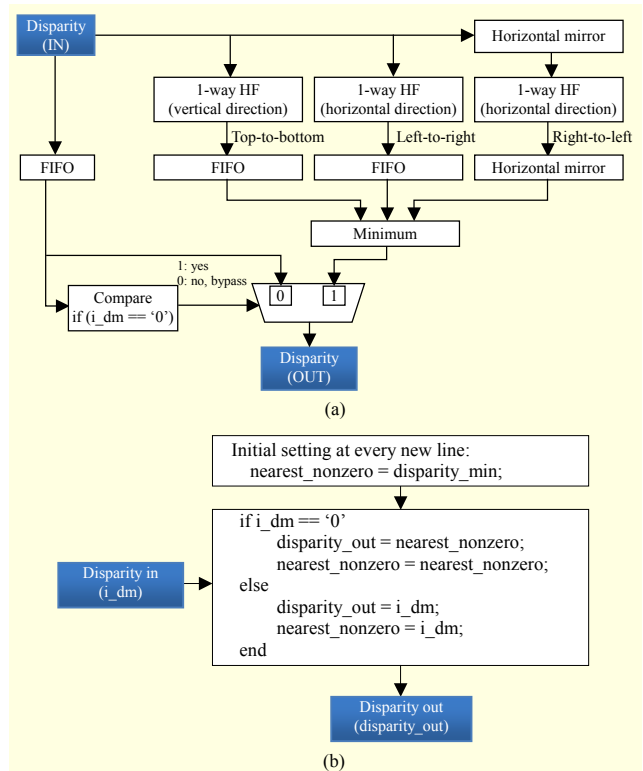


Fig. 12. Proposed scheme for 3-way HF: (a) top scheme for 3-way HF and (b) scheme for 1-way (horizontal/vertical) HF.

note that the clock tick used in the mirroring is also equal to $t(L)$. Then, the path of top-to-bottom is equal to “ $t(L) + t(p) + t(L)$,” that of left-to-top is equal to “ $t(p) + t(L) \times 2$.” Finally, that of right-to-bottom is equal to “ $t(L) + t(p) + t(L)$.”

At the final stage in Fig. 12(a), or the “Compare” stage, when the input data have a hole, whose value is “-1,” the output is the nearest minimum value in the three directions; otherwise, the output is the delayed input data.

2. VC

Equation (16) is a function to calculate the MD in substitution of the variance for a simple hardware implantation. Figure 13 shows the hardware scheme for a VC consisting of the MD and VC logic. An arithmetic mean of input data inside a 9×9 window is calculated by a mean calculator, shown in Fig. 13(b), in the first step. In addition, after calculating the absolute difference between the mean and input data, the MD is calculated by the mean calculator again, which is the arithmetic mean of the absolute difference, as shown in Fig. 13(a). We designed the window block generation to gather the candidate data of the current processing window with buffering, as shown in Fig. 13(c). It should be noted that we do not use a sequential data operation, such as an integral image,

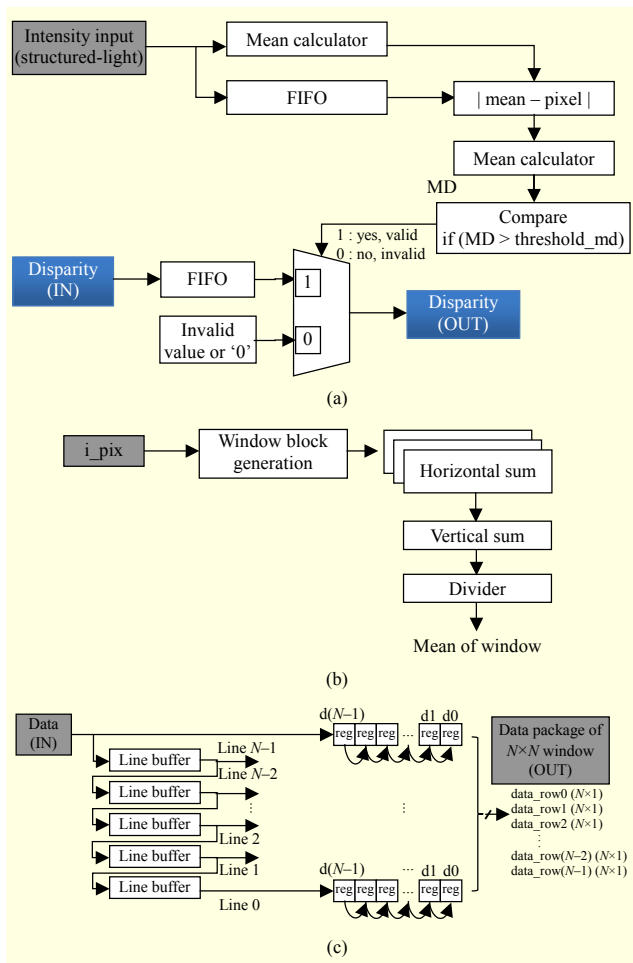


Fig. 13. Proposed scheme for VC with MD: (a) top scheme for VC, (b) scheme for mean calculator, and (c) scheme for block generation (also used in WMF).

because of reusability in the weight-based filter calculations. After obtaining the MD value, it is necessary to compare it against a threshold fixed in advance to conduct a VC.

3. WMF

Fahmy proposed an architecture for a WMF for 1-directional data in [18], and it is easy to expand it for 2-directional data with a simple modification when the weights for the data are invariant. However, when the weight varies pixel by pixel, it is impossible to use the scheme in [18]. Finally, we designed a WMF for data having varying pixel by pixel weights.

We propose a novel architecture consisting of a filter mask calculator and median calculator for a WMF based on a cumulative histogram. Figure 14 shows the novel WMF architecture. Although, in Fig. 14(b), the filter mask calculator has to calculate the exponential function of (17), as a substitution, we use look-up-table ROM having a 16-level weight to reduce the hardware resources. In addition, we found

that this reduction is safe, because there is little difference between the outputs of the full function and look-up-table based on certain experiments. As shown in Figs. 14(c) and 14(d), each bin node consists of an adder and comparator. In addition, there has to be 256 bins for our system because each disparity level needs to have one bin. In Fig. 14(d), a “bin_node_adder” in each bin obtains two inputs, the first being a weight and the second a “inc_en,” which is an incremental sign. In addition, it outputs “sum(node(n)),” which is the sum of the weights having a valid inc_en.

On the other hand, median_location_calculator outputs the med_location, which is half of the sum of all weights. Then, after comparing med_location and sum(node(n)), bin_node_comparator sets med_en(n) to “1” when sum(node(n)) is greater than med_location; otherwise, it is “0.” Lastly, a priority encoder outputs the index (n) of the first bin whose med_en(n) is “1” when being searched in an incremental direction.

VI. FPGA Implementation

Figure 15(b) shows the FPGA system (named TriNet) where the proposed architecture is implemented. It consists of a stereo emulator 15(c) and a stereo head 15(a). In detail, we use the E2V661 CMOS sensor having a good quantum efficiency in the IR band to build the stereo head. The taken stereo image’s resolution is $1,280 \times 720$ pixels (HD resolution) at a rate of 60 fps and has 8-bit gray-scale data per pixel.

In addition, the stereo emulator consists of four FPGAs, DDR3 SDRAM, an application processor having a USB3.0 client function, and a video stream de-serialization module having an HDMI connector. In addition, the stereo head consists of stereo cameras, active projectors, an application processor having a USB3.0 client function, and a video stream serialization module having an HDMI connector.

Figure 16 shows the real-time results of TriNet. It has the functions of image stopping and capturing to make a test bench database. The reduction of the outlier in the disparity map can be noted from Fig. 16(b) through to Fig. 16(e). The proposed design can achieve 60 fps for $1,280 \times 720$ IR stereo images for a 256-level disparity range at a 58 MHz clock speed.

In addition, we have compared the proposed system to a commercial 3D depth sensor to evaluate it not quantitatively but qualitatively. Microsoft’s “Kinect One” (Kinect version 2, emerging in 2014) is considered as the state of the art among the sparse 3D depth sensors emerging up to now. Although Kinect One is based on a time-of-flight method similar to radar, we’re able to compare ours to it because it can provide the depth in image format.

Figure 17 shows the environment used to compare Kinect

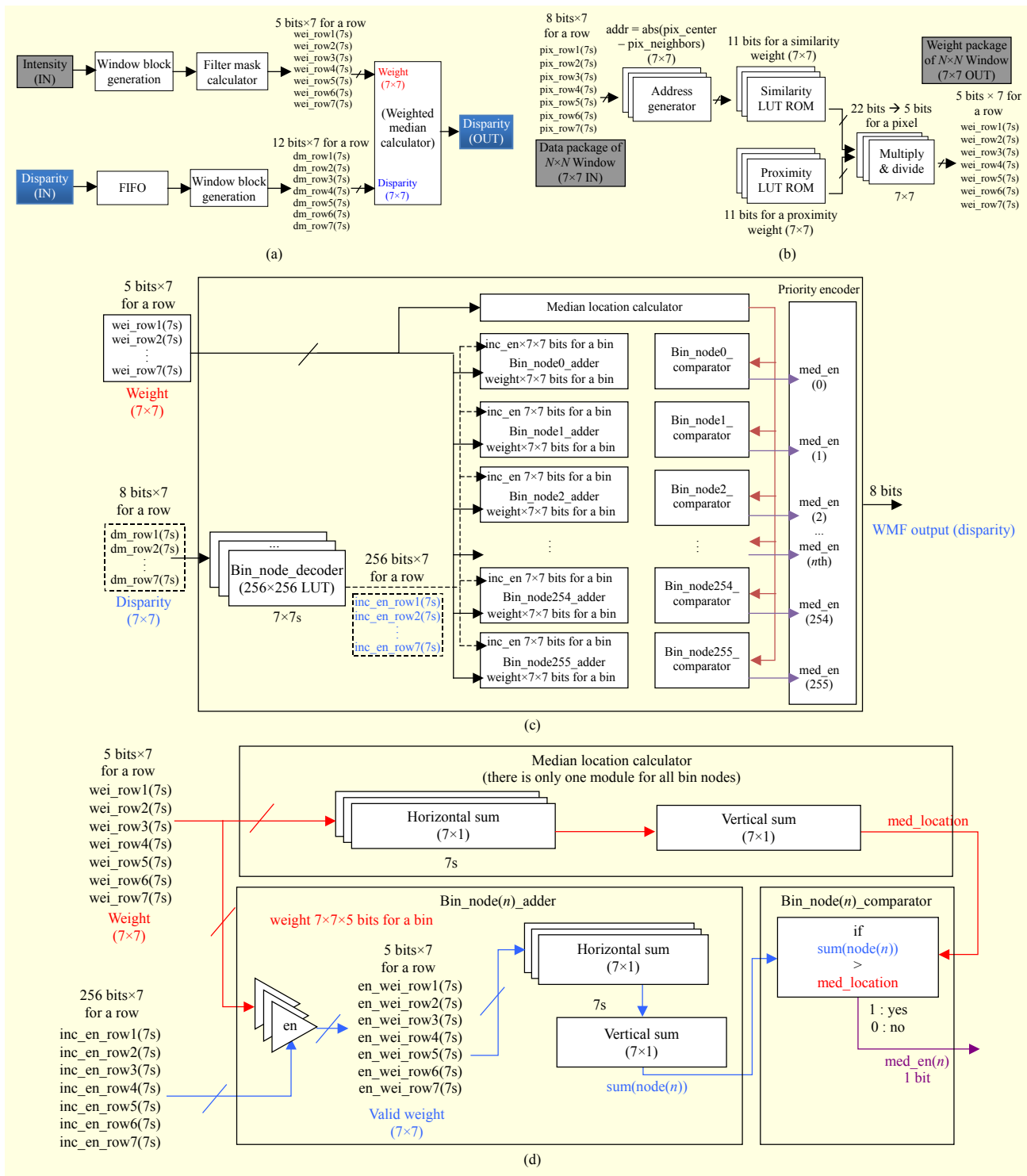


Fig. 14. Proposed scheme for WMF: (a) top scheme for WMF, (b) scheme for filter mask calculator, (c) scheme for median calculation, and (d) scheme for each bin_node (nth bin_node).

1 with the proposed system. In the figure, it can be seen that we positioned Kinect 1, Kinect 2, and TriNet at different distances from subjects so as to give them the same field of view as the subjects; 240 cm, 200 cm, and 300 cm, respectively. And the distance from TriNet to a background wall is 350 cm. Figure 18

shows the results of the three systems — Kinect-1 18(a); Kinect-2 18(b) and 18(d); and the proposed system 18(c) and 18(e) — for various scenes in an indoor environment. Observing the results, one can notice that the shape and edges of 18(b) and 18(c) are sharper than that of 18(a). In addition,

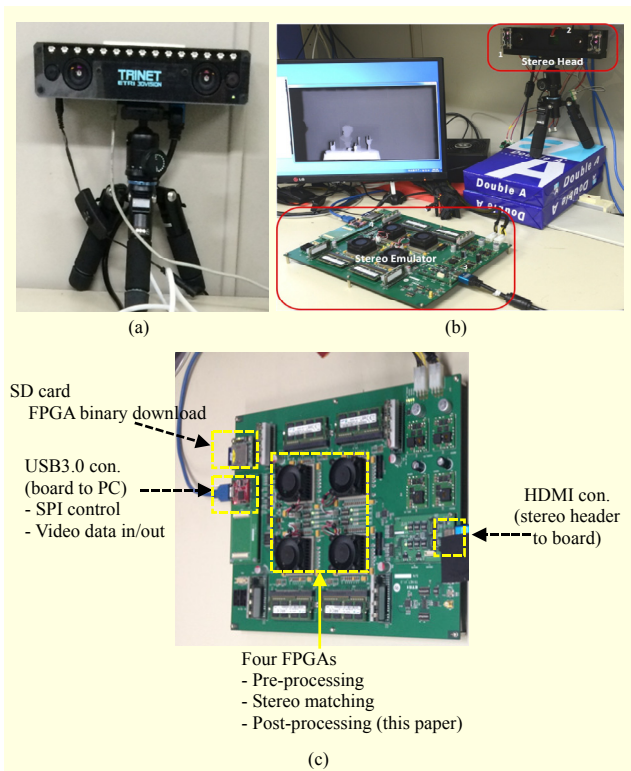


Fig. 15. Stereo vision system (TriNet) processing of proposed algorithms: (a) stereo head, (b) stereo vision system, and (c) stereo emulator (FPGA).

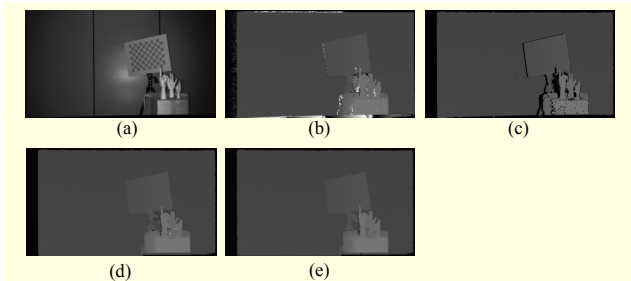


Fig. 16. Real-time FPGA results of proposed post-processing architecture: (a) intensity image (left), (b) disparity map (left ref.), (c) disparity map (LRCC), (d) disparity map (HF), and (e) disparity map (WMF).

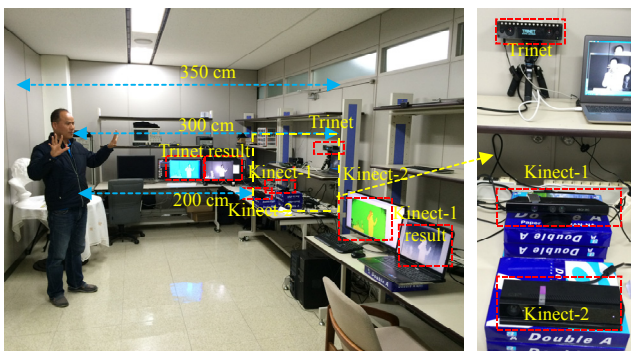


Fig. 17. Environment for comparing TriNet (including proposed algorithm) and commercial Kinect series (ver. 1, ver. 2).

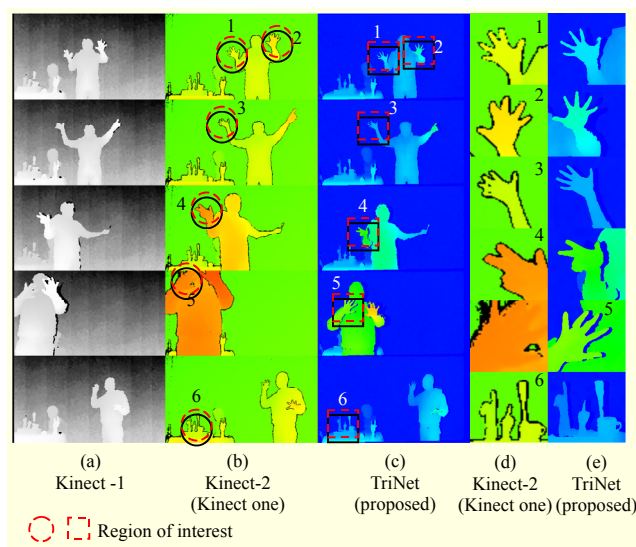


Fig. 18. Results of comparing TriNet (including proposed algorithm) and commercial Kinect series (ver. 1, ver. 2).

Table 5. Usage of FPGA for proposed architecture (XC7V2000TFLG1925-SP2).

Block name	Slice (305,400)		Memory (36 kbits)
LRCC	336	0.11%	1
HF	508	0.16%	10
VC	1,377	0.45%	9
WMF	37,534	12.29%	10
Total	39,755	13.01%	30 (135 Kbyte)

for the region of interest, the results from the proposed system (18(e)) are more precise and less blurred than those from Kinect-2 (18(d)). Moreover, the depth error between the real distance (measured by telemeter) and our measured distance is below 1 cm for a subject near the center of an image at a distance of 3 m, except in occlusion or textureless regions.

Table 5 describes the resources (slice and memory) used in the FPGA for implementing the proposed architecture. Because the weight calculation needs a lot of multiplication operations and look-up tables, it should be noted that the WMF uses a larger proportion of the total system resources than any other type of logic. Accordingly, we used 13.01% of slices at Xilinx Virtex®-7 to implement the proposed post-processing algorithm.

Lastly, Table 6 shows a comparison of the specifications among other reported researches on real-time stereo vision implementation. According to the table, the proposed system has competitive performance in comparison to the other reported researches. In addition, because we use both passive and active stereo at the same time, the system is robust in

Table 6. Reported stereo vision systems to date.

Name of reported stereo vision system	Image size	Disparity range	Rectification	Hz	Projection method
DeepSea ASIC (2004) [21]	512 × 480	52	Firmware	200	Passive
MSVM-III FPGA (2004) [22]	640 × 480	64	No	30	Passive
Jin's FPGA (2010) [20]	640 × 480	64	FPGA	230	Passive
SGM FPGA (2010) [23]	640 × 480	128	FPGA	30	Passive
Jeon's GPU (2013) [11]	400 × 300	64	No	24	Passive
This paper (TriNet)	1,280 × 720	256	FPGA	60	Hybrid

various illuminations.

VII. Conclusion

We present a novel post-processing algorithm and its VLSI architecture for a high-quality depth map in hybrid active stereo vision. In particular, the proposed system simultaneously uses passive and active stereo vision information to improve the reliability of the three-dimensional disparity in a hybrid stereo vision system. The proposed architecture consists of four hardware-optimized sub-blocks in a cascade manner; that is, consistency checking, HF, VC, and WMF. In particular, the proposed real-time, compact semi-2D HF method uses less resources than the 8-way method to fill holes indicating inaccurate depth values caused by mismatching at occlusion regions. In addition, the proposed tiny VC logic using an MD in substitution of the variance for reducing resources simultaneously uses the active pattern and passive object scenes for improving performance in searching textureless regions. Lastly, a novel architecture, 2D WMF, uses the similarity and proximity weight of a passive scene and disparity of a hybrid matching to reduce outlier noise occurring from mismatching.

The proposed architecture implemented on a single FPGA, where only 13.01% of slices of a XC7V2000TFLG1925 are used, can achieve 60 fps for stereo images having 1,280 × 720 resolution. In addition, it has a 256-level disparity range. Although the proposed algorithm is for the disparity of *active* stereo vision, it somehow shows a good performance for the Middlebury stereo image sets, which are datasets for *passive* stereo vision. The experimental results show that the error rate of the proposed algorithm (8.30%) is less than that of the raw disparity (13.7%) for these datasets. In addition, the error rate decreases from 10.12% to 5.77% for a real-world dataset.

Finally, we present the output images from Microsoft's Kinect-1 and our system for the same scene to compare their results qualitatively. In these comparing experiments, the results from the proposed system are more precise and less blurred than those from Kinect-1 and Kinect-2. In addition, the depth error is below 1 cm for a subject near the center of an image at a distance of 3 m, except for occlusion or textureless regions.

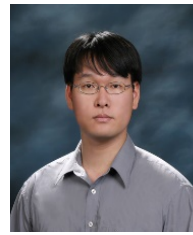
References

- [1] J.-C. Jeong et al., "High-Quality Stereo Depth Map Generation Using Infrared Pattern Projection," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 1011–1020.
- [2] J.H. Chang, J.C. Jeong, and D.-H. Hwang, "High-Quality Stereo Depth Map Generation Using Infrared Pattern Projection," *Proc. British Mach. Vis. Conf.*, Nottingham, UK, Sept. 2014.
- [3] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jan. 4–7, 1998, pp. 839–846.
- [4] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, Apr. 2002, pp. 7–42.
- [5] J. Heikkila and O. Silven, "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, San Juan, Puerto Rico, June 17–19, 1997, pp. 1106–1112.
- [6] D.I. Han et al., "The Design of HD Image Rectification Architecture Using Floating Point IP," *IERI Procedia*, Saipan, USA, vol. 6, 2014, pp. 39–44.
- [7] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," *Proc. ECCV*, Stockholm, Sweden, May 2–6, 1994, pp. 151–158.
- [8] A. Hosni, M. Gelautz, and M. Bleyer, "Accuracy-Efficiency Evaluation of Adaptive Support Weight Techniques for Local Stereo Matching," *DAGM OAGM Symp.*, Graz, Australia, vol. 7476, Aug. 28–31, 2012, pp. 337–346.
- [9] F. Tombari et al., "Classification and Evaluation of Cost Aggregation Methods for Stereo Correspondence," *Proc. IEEE Comput. Vis. Pattern Recogn.*, Anchorage, AK, USA, June 23–28, 2008, pp. 1–8.
- [10] C. Çigla and A.A. Alatan, "Efficient Edge-Preserving Stereo Matching," *IEEE Int. Conf. Comput. Vis. Workshops*, Barcelona, Spain, Nov. 6–13, 2011, pp. 696–699.
- [11] C.C. Pham and J.W. Jeon, "Domain Transformation-Based Efficient Cost Aggregation for Local Stereo Matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, Oct. 2012, pp. 1119–1130.
- [12] S.M. Choi et al., "Post-Processing Algorithms for Real-Time Active Stereo Vision," *IEEE Int. Symp. Consum. Electron.*, Jeju, Rep. of Korea, June 22–25, 2014, pp. 1–2.

- [13] S.M. Choi et al., "A FPGA Based Real-Time Post-Processing Architecture for Active Stereo Vision," *IEEE Int. Symp. Consum. Electron.*, Jeju, Rep. of Korea, June 22–25, 2014, pp. 1–2.
- [14] S.B. Kang, R. Szeliski, and J. Chai, "Handling Occlusions in Dense Multi-view Stereo," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, Kauai, HI, USA, vol. 1, Dec. 8–14, 2001, pp. 103–110.
- [15] L. Wang et al., "Stereoscopic Inpainting: Joint Color and Depth Completion from Stereo Images," *IEEE Comput. Vis. Pattern Recogn.*, Anchorage, AK, USA, June 23–28, 2008, pp. 1–8.
- [16] H. Jiang, R. Gao, and X. Liu, "Research of Stereo Matching Based on Improved Median Filter," *Int. Conf. Electr. Electron.*, Nanchang, China, vol. 4, June 20–22, 2011, pp. 479–486.
- [17] L. Yin et al., "Weighted Median Filters: A Tutorial," *IEEE Trans. Circuits Syst. II: Analog Digital Signal Process.*, vol. 43, no. 3, Mar. 1996, pp. 157–192.
- [18] S.A. Fahmy, P.Y.K. Cheung, and W. Luk, "Novel FPGA-Based Implementation of Median and Weighted Median Filters for Image Processing," *Int. Conf. Field Programmable Logic Appl.*, Tampere, Finland, Aug. 24–26, 2005, pp. 142–147.
- [19] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, Madison, WI, USA, no. 1, June 18–20, 2003, pp. 195–202.
- [20] 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, Jan. 2010, pp. 15–26.
- [21] J.I. Woodfill, G. Gordon, and R. Buck, "Tyzx DeepSea High Speed Stereo Vision System," *Conf. Comput. Vis. Pattern Recogn.*, Washington, DC, USA, 2004, pp. 41–46.
- [22] Y. Jia et al., "A Miniature Stereo Vision Machine (MSVM-III) for Dense Disparity Mapping," *Proc. Int. Conf. Pattern Recogn.*, Cambridge, UK, Aug. 23–26, 2004, pp. 728–731.
- [23] C. Banz et al., "Real-Time Stereo Vision System Using Semi-global Matching Disparity Estimation: Architecture and FPGA-Implementation," *Int. Conf. Embedded Comput. Syst.*, Samos, Greece, July 19–22, 2010, pp. 93–101.



Seungmin Choi received his BS degree in electronics engineering from Chung-Ang University, Seoul, Rep. of Korea, in 2002 and his MS degree in electrical and computer engineering from Seoul National University, Rep. of Korea, in 2004. He has been a senior researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, since 2004. His research interests include VLSI/ASIC chip design for image processing, stereo vision, computer vision, and medical imaging systems.



Jae-Chan Jeong received his BS degree in computer engineering from the Korea University of Technology and Education, Cheonan, Rep. of Korea, in 2006 and his MS and PhD degrees in computer engineering from the University of Science and Technology, Daejeon, Rep. of Korea, in 2009 and 2015, respectively. He is a senior researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His main research interests include robotics, stereo matching, and computer vision.



Jiho Chang received his BS and MS degrees in electronics engineering from Chung-Ang University, Seoul, Rep. of Korea, in 2004 and 2006, respectively. He is currently working as a senior researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His main research interests include stereo matching, computer vision, and FPGA design technology.



Hochul Shin received his MS and PhD degrees in mechanical engineering from KAIST, Daejeon, Rep. of Korea, in 1999 and 2005, respectively. He has been with the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea as a full-time senior researcher, since 2005. Additionally, since 2006, he has been an affiliated professor at the University of Science and Technology, Daejeon, Rep. of Korea. His research interests include visual recognition of intelligent robots, robotic surgery, and biomechanics.



Eul-Gyoon Lim received his BS and MS degrees in mechanical engineering from Yonsei University, Seoul, Rep. of Korea, in 1998 and 2000, respectively. He worked at Samsung Electronics from 2000 to 2001 as a member of the engineering staff. Since 2001, he has worked at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea as a principal researcher. His research interests are in stereo matching algorithms and their implementation.



Jae Il Cho received his BS and MS degrees in electrical engineering from Sungkyunkwan University, Suwon, Rep. of Korea, in 1990 and 1992, respectively. Since joining the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea in 1992, he has carried out research on communication networks and intelligent service robot systems. His research interests include robotics, pattern recognition, computer vision, and ASIC design.



Daehwan Hwang received his PhD degree in electrical engineering from Sungkyunkwan University, Suwon, Rep. of Korea, in 1998. Since joining the Electronics and Telecommunication Research Institute, Daejeon, Rep. of Korea in 1990, he has carried out research on ISDN, ATM, IP multimedia networking, system on a chip, cognition systems, and 3D sensors. His research interests include audio-visual digital signal processing, multimedia devices, multimodal sensors, and ASIC/SoC design architecture.