Reliable Stereo Matching With Depth Measurement in Video Frames Using FPGA

J. Subi
Research Scholar, Nandha Engineering College, Erode, India.

S. Nandhini
Assistant Professor, Nandha Engineering College, Erode, India.

Abstract – The emerging market of digital 3-D film productions in HD resolution leads to the need for high-quality equipment in the production chain. First, the images are treated in order to improve their pixel to pixel correspondence and reduce illumination differences. After that, stereo matching is addressed using different methods with emphasis on local ones like the sum of absolute distances and normalized cross correlation. A low-complexity depth-reliability-based stereo matching algorithm and an efficient scanline memory-merging implementation scheme are proposed. A bidirectional depth propagation flow is then adopted to fill the unreliable segments by using reliable information. Moreover, a set of predefined function-specific reliability variables are extracted to further improve depth quality in the occluded and smooth regions, which can reduce 39% bad pixels obtained by applying the basic 7 × 7 window-based matching. The proposed scanline memory-merging scheme along with data prefetching can lead to 32.7% savings on the scanline memory area and relax the requirements of external frame buffer size and bandwidth.

Index Terms – Reliability-based computation, stereo-matching, window-based sum of absolute differences (SAD) architecture.

1. INTRODUCTION

Stereo matching is a classic research topic in computer vision, with applications in multi view technology, 3-D films, and 3-D virtual reality. Although matching algorithms have been developed for several decades, few studies have considered cost-efficient, real-time very large- scale integration (VLSI) architecture design for such algorithms. Traditional stereo-matching algorithms can be divided into global optimization approaches [1]–[3] and local window based approaches [4]–[8]. An extensive complexity and quality comparisons of global and local optimization algorithms were reported in [4]. Global optimization provides high depth quality because of its energy optimization functions, but its computation load may be impractical for hardware implementation. Many local window-based stereo-matching algorithms use either a variable window size or an irregular window shape [5], [6] to deal with depth discontinuity regions. Different cost aggregation [7]–[10], post processing [11], and preprocessing [12] techniques are also developed to enhance the depth quality of local window-based methods. Generally, the main hardware cost of local window-based stereo-matching algorithms is dominated by the aggregation window size. Scharstein et al. [4] investigated the resulting depth quality with respect to various window sizes and demonstrated that the quality is unacceptable if the window size is smaller than 13 × 13. A relatively large window size is usually chosen for obtaining reliable matches, which leads to high complexity and memory cost for hardware design, especially for irregular window shapes. This paper proposes a depth reliability- based stereo-matching algorithm that can significantly improve the depth quality obtained by adopting small window sizes (e.g., 7×7) in conventional solutions. Moreover, a cost-effective scanline based VLSI hardware architecture are developed for stereo-matching. The proposed algorithm and hardware architecture are applicable for different requirements of quality and complexity tradeoff.

Several stereo-matching algorithms have been developed for VLSI hardware architectures. In [7], a minicensus adaptive support weight disparity estimation algorithm was proposed on the basis of the adaptive support weight (ADSW) [8]. A good depth quality is obtained, but the memory and bandwidth requirements are high although some simplification and bandwidth reduction techniques are adopted. In [13] and [14], a simple sum of absolute differences (SAD) matching technique was adopted, but it does not perform well in occluded and smooth regions. The occlusion problem was addressed in [15] by adopting left-right consistency (LRC); however, the computational complexity is doubled accordingly. In [16], a stereo system based on a phase-based stereo-matching algorithm called local weighted phase-correlation was proposed. Although the method is robust to modest deformations and smooth lighting variations between corresponding images, the depth quality is relatively poor. A segmentation based design [17] with ADSW was proposed to reduce the required hardware resources by using a small window size (13×13). However, the depth quality suffers because of poor segmentation results.

A cost-effective stereo-matching algorithm along with its VLSI architecture design and optimization are proposed in this paper. Instead of modifying the window shape and the weighting for cost aggregation, matching reliability is extracted and only reliable depths are used to improve depth quality. An efficient depth propagation flow is then proposed to fill unreliable regions by using reliable results. The developed algorithm improves the quality of stereo matching by using a small window (e.g., 7 × 7). For reliable depth decision, a simple rule is first used to identify the most reliable depths from the trivial depth generated using the conventional regular square window. Then, a smooth region detector is introduced to filter out unreliable depths in smooth regions by investigating the variation of the cost value distribution within the search range. Occluded regions are excluded from the determination of reliable depths by using a local method that checks the depth difference of neighboring pixels and a global approach that references the depth histogram. Finally, constrained reliable depth dilation is applied to extend the reliable depth according to the depth and intensity similarities of neighboring pixels. For reliable depth propagation, segments consisting of continuous unreliable depths within a single scanline are categorized into three types and replaced with reliable depth at the segment ends. Moreover, horizontal and vertical propagation schemes are employed by taking into account the fattening effect and inter scanline inconsistency. Although the algorithm in [18] also addressed the concept of propagating the reliable depth, its line segment construction is irregular and the seed pixel detection includes a left–right check, making hardware realization expensive.

For hardware architecture optimization, a memory merging and relative data prefetching scheme is developed in this paper to reduce the scanline buffer area by 32.7%. The search direction of SAD computation and data scheduling are managed to minimize data access latency. The processing elements (PEs) for parallel SAD computation are properly shared, leading to a PE usage reduction of about 73%. The implementation results demonstrate that the proposed design can achieve high area efficiency with performance competitive with that of existing algorithms with hardware implementations. For example, the area efficiency of the proposed design is 11 times better than that in [7].

The rest of this paper is organized as follows. Section II introduces several local window-based stereo-matching algorithms. The proposed reliability-based algorithm and its optimized hardware architecture are described in Sections III and IV, respectively. Experimental results are presented and compared in Section V. Section VI concludes this paper.

2. RELATED WORK

Stereo-matching algorithms can be divided into two categories: global optimization approaches and local window based approaches.

2.1. Global Opti\ ncbxdt mization Approaches

Global optimization approaches such as dynamic programming (DP) [1], graph cuts [2], and belief propagation [3] obtain good depth accuracy because of their energy optimization functions, which can be expressed as

$$\text{E (d)= Edata (d)} + \alpha\text{Esmooth (d)}$$

$$\text{Edata (d)=}\sum\nolimits_{(x,y)} \mathrm{C}(x, y, d(x, y))$$

where the data term Edata(d) measures the correlation or similarity of two image pairs at a particular disparity d, the smoothing term Esmooth(d) is the smoothness of depth,α is the weighting factor between them, (x, y) is a pixel’s coordinates, d(x, y) is the matching correspondence in the reference image, and C(x, y, d(x, y)) is the cost calculation. The smoothing term often measures the differences between neighboring pixels’ disparities

$$\text{Esmooth(d)=}\sum\nolimits_{(x,y)} \begin{pmatrix} -p\big(d(x,y) - d(x+1,y)\big) + \\ p\big(d(x,y) - d(x,y+1)\big) \end{pmatrix}$$

Where ρ is a monotonically increasing function of the disparity difference [4]. Global optimization often achieves better depth quality than that obtained with local window-based approaches at the cost of high computational complexity and storage requirements, which may be impractical for VLSI implementation. In contrast, local stereo-matching algorithms are more suitable for hardware realization and are more likely to meet the demands of real-time computation, though they usually sacrifice depth quality.

2.2. Local Window-Based Approaches

The development of local window-based algorithms focus on the matching cost computation and aggregation. Specifically, such algorithms choose their local window sizes and shapes, aggregate the matching costs according to the defined cost function and determine the final disparity using a local winner take all (WTA) optimization at each point.

The simplest local window-based method matches the corresponding pixels with a fixed square window. The method assumes that the pixels in the window have similar depth; therefore, estimation error occurs when the window shifts to a depth discontinuity. The algorithm in [4] uses a shifting window to deal with depth discontinuity. When the window goes through a depth discontinuity, the location of the window that minimize the matching cost is selected. Several studies have proposed modifying the shape of the window for properly handling depth discontinuity, such as those using a compact window [5] and an adaptive binary window [6]. The compact window algorithm chooses a window size and shape by optimizing over a large class of compact windows via a minimum ratio cycle. The window cost function is denoted as

$$E\,(W) = \frac{\sum_{q\in W} \operatorname{err}(q,d)) + \sum_{e\in C_W} b(e,d)}{\sum_{q\in C_W} n(q,d)}$$

Where err(q, d) is the error measurement for pixel q with disparity d [similar to the absolute difference (AD)], CW is the corresponding cycle, e is the edge of CW, b(e, d) is the edge-dependent factor, and n(q, d) is the normalization term. The minimum ratio cycle is used to find the window with the lowest matching cost. The boundaries of compact windows are located at edges of the image content. The adaptive binary window is used to segment regions with a given depth, and thus the window shape is irregular. To construct the support matching window, the color similarity of pixels in a region (e.g., a 33 × 33 window) is used to build the window. Pixels with a small Euclidean distance are grouped to form the window shape in the CIELab color space. Because the algorithms proposed in [5] and [6] employ irregular window shapes, they can handle depth discontinuity better than can other algorithms. However, a hardware implementation of algorithms on the basis of irregular window shapes is challenging. The ADSW approach [8] adjusts the support weight of the pixels in a given support window according to the color similarity and geometric proximity to reduce image ambiguity. The support-weight is computed on the basis of Gestalt principles and can be written as

$$W(p,q) = \exp\left(-\left(\frac{\Delta c_{\mathrm{pq}}}{\gamma_c}, \frac{\Delta g_{\mathrm{pq}}}{\gamma_p}\right)\right)$$

Where _cpq and _gpq are the color difference and spatial distance between pixels p and q, respectively, γc is a constant factor 14, and γp is related to the field-of-view of the human visual system. The dissimilarity between pixels is measured using

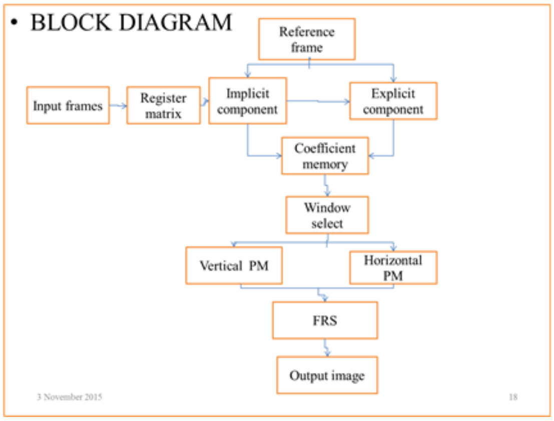
$$E(p,\overline{p_d}) = \frac{\sum_{q\in N_p,\overline{q_d}\in N_{\overline{p_d}}} W(p,q)W(\overline{p_d},\overline{q_d})e(q,\overline{q_d})}{\sum_{q\in N_p,\overline{q_d}\in N_{\overline{p_d}}} W(p,q)W(\overline{p_d},\overline{q_d})}$$

Where pd and qd are the corresponding pixels in the target image when the pixels p and q in the reference image have a disparity value d and e(q, qd) is the truncated AD. This approach implicitly segments regions with similar color, similar to the schemes presented in [5] and [6] but with better performance. However, increasing the window size.

As discussed previously, fixed window-based approaches are better choices for real-time hardware implementations. However, existing local methods demand a large window size (i.e., 21 × 21 to 33 × 33) and a large number of scanline buffers for hardware realization with the real-time constraint. The proposed stereo-matching algorithm considers the reliability of depth decisions by using a relatively small window (e.g., 7 × 7) to reduce the hardware cost while maintaining similar depth quality. (e.g., a 33 × 33 window) increases the required hardware resources and an irregular window shape is difficult to implement in hardware.

3. COST-EFFECTIVERELIABILITY-BASED STEREO-MATCHING ALGORITHM

In most stereo-matching algorithms, it is difficult to acquire reliable results utilizing low-complexity local window-based methods only, especially for cases with repeated patterns, smooth regions, and occluded regions. In addition, hardware constraints such as limited storage restrict the available matching window size. A depth-reliability-based stereo-matching algorithm is proposed to generate a good depth result even for difficult cases. Note that the required matching window size is smaller than those of existing window-based approaches with similar depth quality. Therefore, the proposed algorithm consumes very limited resources and is thus suitable for VLSI realization. The algorithm can be divided into three stages: the SAD phase, the depth reliability computation phase, and the reliable depth propagation phase, as shown in Fig. 1. The SAD phase consists of a simple small window-based cost computation performed using a WTA decision unit. The results, containing many mismatches, are examined in the reliability computation phase. Poor decisions are removed and reliable depth values are extended to fill unreliable regions in the depth propagation stage.



- BLOCK DIAGRAM

Reference frame

Input frames

Register matrix

Implicit component

Explicit component

Coefficient memory

Window select

Vertical PM

Horizontal PM

FRS

Output image

3 November 2015

18

Fig 1 Stereo matching algorithm flow

3.1. Fast Initial Reliability Decision

Conventional window-based algorithms require a large matching window to guarantee performance. The depth results derived using a small window, such as 7×7 in this paper, are often not reliable; thus, it is crucial to identify reliable depths from the derived results. In this paper, depth reliability is initially extracted from the distribution properties of computed SAD values within the search range.

Local WTA is a well-known method for determining the final disparities in window-based stereo-matching algorithms. The major problem of this approach is that the minimum cost is not always reliable. To relax this problem and search for more reliable depths, the distribution of the matching cost within the search range is first examined. As shown in Fig. 2(b) and (d), there exists a distinct minimum cost within the search range, and thus the matching decision is correct (i.e.,the depth is more reliable). In contrast, the matching decision is more risky for the example shown in Fig. 2(c) and (e) as there is no distinct minimum cost. To distinguish these two cases, the difference between the minimum and the second minimum matching costs is considered. However, a misjudgment might occur when the second minimum cost is located in the neighborhood of the minimum cost. Under such circumstances, the depth should be treated as reliable information because the positions of the minimum and the second minimum values are close.

Taking into account the preceding two observations, the minimum dm1 and the second minimum dm2 in the search range are recorded and then the initial depth reliability Rinit(x) is determined as

$$R_{\text{init}}(x) = \begin{cases} 1, & \dfrac{C(x,d_{m2}) - C(x,d_{m2})}{C(x,d_{m2})} > T_c \\ 0, & \text{otherwise} \end{cases}$$

$$d_{m1} = \arg\min_{d \epsilon S} C(x,d)$$

$$d_{m2} = \arg \min_{d \epsilon S \cap |d-d_1| > T_d} C(x,d)$$

Where x is the horizontal coordinate, C (x, d) is the cost value of disparity d within the search range **S**, Tc is the normalized threshold for determining if the depth is reliable, and Td is the disparity tolerance for the second minimum cost selection. Specifically, the initial reliable depth is estimated using a low complexity operation, which is the difference between the minimum (C (dm1)) and the second minimum (C (dm2)) cost values normalized by the minimum cost value. The second minimum cost is selected outside a tolerance range from the minimum value, that is, dm2 is defined in the depth search range excluding the neighborhood of dm1.

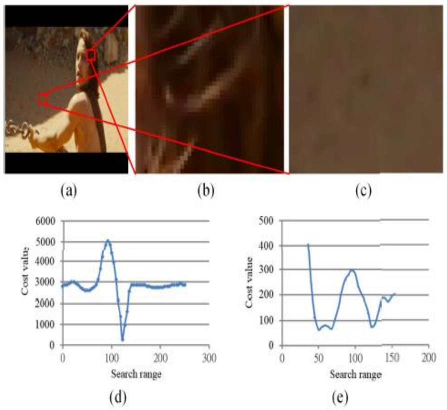


Fig. 2. Reliability of matching cost. (a) Original image. (b) Enlarged texture region. (c) Enlarged smooth region. (d) Distribution of matching cost in texture region. (e) Distribution of matching cost in smooth region.

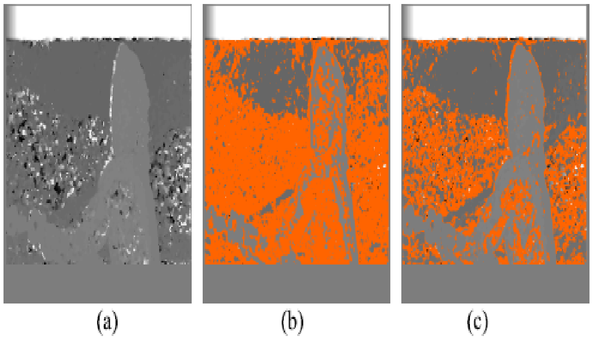


Fig. 3. (a) Trivial depth. (b) Initial reliable depth (orange: unreliable). (c) Initial reliable depth considering minimum cost tolerance.

Fig. 3(a) and (b) shows the trivial depth generated from the conventional 7 × 7 window-matching algorithm and the generated initial reliable depth, excluding the region marked in orange, without considering the tolerance range. The initial reliable depth decision obtained using (6) is shown in Fig. 3(c) in which plenty of reliable depths are recovered. The proposed initial reliability decision approach approximates the reliability values by using low-complexity computation.

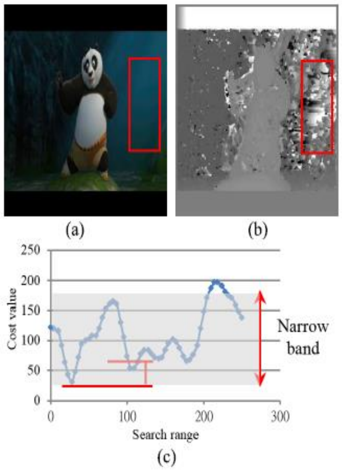


Fig. 4. Matching ambiguity problem in low-texture region. (a) Original image. (b) Trivial depth. (c) Distribution of matching cost in low-texture region.

3.2. Deterministic Unreliable Depth Identification

The initial reliable depth information is further refined by carefully identifying possible low-texture and occlusion regions. Specifically, those reliable depths determined in the initial reliability decision step are changed to be unreliable if they are located in the low-texture and occlusion regions. Most stereo-matching algorithms suffer from bad performance in such regions. Smooth and occlusion regions are detected efficiently in the deterministic unreliable depth identification phase.

3.2.1. Minimum Cost Ambiguity Problem in Low-

Texture Regions: A smooth region is defined as a large area without texture, as the marked regions shown in Fig. 4(a) and (b). In smooth regions, the SAD costs computed using a small matching window within a limited search range are similar; thus, the correct matching pair cannot be detected. This implies that the matching decision in smooth regions should be regarded as unreliable. The simple initial depth reliability decision rule in (6) may not distinguish all the distribution cases of matching costs appropriately in smooth regions. From our observation, the matching costs in the search range of smooth regions are usually distributed in a narrow band, as shown in Fig. 4(c). Thus (7) is adopted to detect such a distribution

$$R_{\text{smth}}(x) = \begin{cases} 1, & |\text{num}(|C(x,d) - C(x,d_{m1})| < T_{\text{nb}}) < T_s \\ 0, & \text{otherwise} \end{cases}$$

Where the function num(d) is the number of disparities d that meet the statement and Tnb is the predefined width of the narrow band for smooth region detection. If the number of cost values close to the minimum cost is larger than Ts, the matching is treated as being performed in a smooth region and regarded as unreliable. The rectified outcomes are shown in Fig. 5, in which most of the reliable depths in low-texture regions determined in the initial reliability decision step are excluded by the low-texture detector.

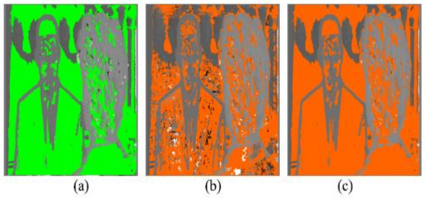


Fig. 5. Unreliable depth elimination in low-texture region. (a) Low-texture region (green). (b) Initial reliability decision. (c) Results of applying low-texture filter.

3.2.2. Occlusion-Induced Visual Fatigue Detection:

Another common challenge is stereo matching in occluded regions (i.e., no true matching pair exists). As shown in Fig. 6, there is no matching pair because the occlusion highlighted in the left image is not visible in the right image. Most stereo-matching algorithms do not consider this issue, thus producing errors in occluded regions. A well-known method for dealing with occlusion is the LRC check [17], which creates two disparity maps (i.e., left to right and right to left) and examines the consistency between them. Although LRC is useful for occlusion detection, it doubles the disparity computation complexity. In [19], occluded regions are detected by considering the motion estimation for more than two images (i.e., a video sequence), and thus frame buffers are required in hardware systems.

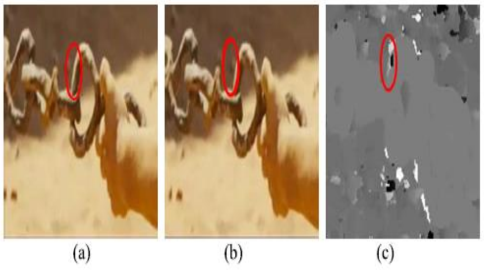
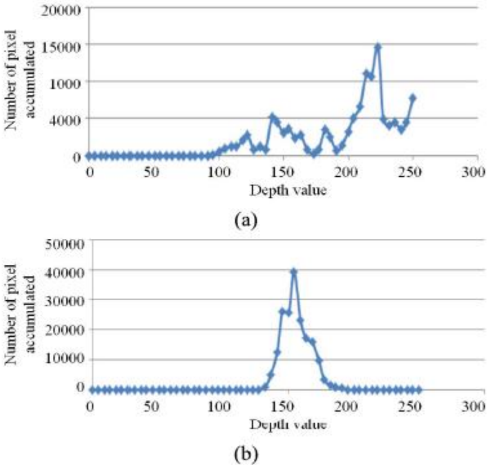


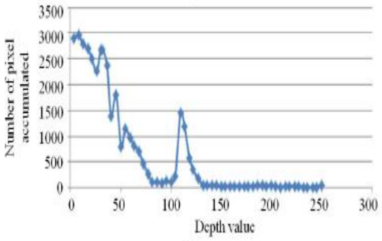
Fig. 6. Occlusion problem in stereo matching. (a) Left image. (b) Right image. (c) Trivial depth with occlusion-induced visual fatigue.

To have a cost-effective solution, the proposed method only detects wrong reliability decisions affected by occlusion, instead of detecting all occluded areas, as done in [19].Occlusion-induced depth error, shown in Fig. 6(c), appears as an unusually large depth difference in the neighborhood of occluded regions owing to inaccurate correspondence. In this paper, a global occlusion filter that uses the reliable depth histogram and a local occlusion filter that considers the depth gradient are developed to remove reliable depths that were potentially generated from occluded regions.

The three general types of derived reliable depth distribution are shown in Fig. 7, in which most of the reliable depths are in the main lobe of the distribution. Reliable depths residing outside the main lobe of the distribution are filtered out by the global filter, because the computed depth values in the occluded regions are usually very different from those in other regions. Fig. 8 shows an example with refined reliable depths in occluded regions obtained by tracing the main lobe of the reliable depth distribution.

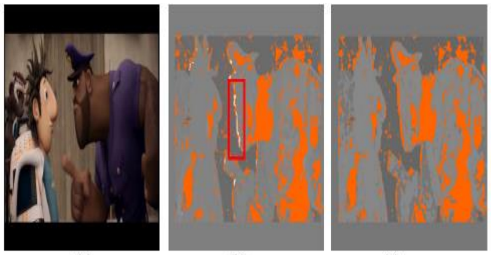


(b)



(c)

Fig. 7. Reliable depth distribution categories. (a) Foreground-dominated. (b) Normal distribution. (c) Background-dominated.



(a) (b) (c)

Fig. 8. Example of applying global occlusion filter. (a) Original image. (b) Depth with occlusion-induced error. (c) Refined reliable depth.

The process can be formulated as

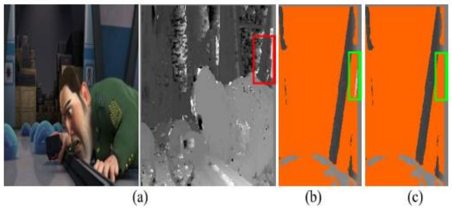
$$R_{\text{gocc}}(x) = \begin{cases} 1, & b_1 < d_t(x) < b_u \\ 0, & \text{otherwise} \end{cases}$$

$$b_l = \arg \min_{0 \le b \le 255} \left(\left| \sum_{i=0}^{b} h[i] - T_{\text{sr}} \right| \right)$$

$$b_u = \arg \min_{b_l b \le 255} \left(\left| \sum_{i=b_l}^{255} h[j] - T_{\text{sr}} \right| \right)$$

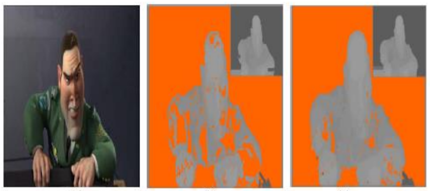
where dt (x) is the trivial depth value of a pixel at horizontal coordinate x, bl and bu are the lower and upper bounds of the main distribution in the reliable depth histogram, respectively, h[i] is the accumulated pixel number of depth value i, and Tsr is the threshold value used to denote the number of reliable pixels that are considered residing outside the main lobe. The value is estimated by the number of reliable pixels multiplied by a predefined fraction, which is set to 2% in our experiments.

Because it is still possible that the reliable depths generated from occluded regions are within the main lobe of the distribution, a local occlusion filter is applied. The filter is basically a local disparity gradient detector applied to detect large reliable depth differences between neighboring pixels. Extensive simulations of test cases indicate that the wrong depth in occluded regions frequently appears as a large depth difference between neighboring pixels. To identify this condition, the x coordinate of the nearest pixel within a specific detection range (e.g., 8 pixels) with reliable depth is obtained as xnr if it exists, and then (9) is used to determine the reliability considering local occlusion.



(a) (b) (c)

Fig. 9. Example of applying local occlusion filter. (a) Original image (left) and its trivial depth (right). (b) Depth with occlusion-induced error. (c) Refined reliable depth.



(a) (b) (c)

Fig. 10. Reliable depth dilation for object integrity. (a) Original image and results (b) without and (c) with refined depth dilation.

$$R_{\text{locc}}(x) = \begin{cases} 1, & \text{otherwise} \\ 0, & \text{abs}|d_t(x - x_{\text{nr}}) - d_t(x)| > T_{ld} \end{cases}$$

Where Tld is a chosen threshold value for denoting a large depth difference between two nearby pixels, which is assumed to rarely appear in natural scenes. Fig. 9 shows the result of applying the local filter to refine the reliable depths. The final reliable depth is determined by considering all of the reliability terms

$$R(x) = R_{\text{init}}(x) \times R_{\text{smth}}(x) \times R_{\text{gocc}}(x) \times R_{\text{locc}}(x)$$

3.3. Constrained Reliable Depth Dilation for Object Integrity

Owing to the strict constraints of the filters described in Section III-B, some points with reliable depths are misjudged as having unreliable depths and a sparse reliable depth map is usually generated, as shown in Fig. 10(b). Because the sparse map is a conservative outcome, constrained reliable depth dilation is then employed to recover some misjudged reliable depths and preserve object integrity.

The proposed dilation method uses proximity and similarity concepts on the basis of Gestalt principles. According to Gestalt principles, reliable pixels are dilated into neighbors (i.e., proximity) if they have similar depths (i.e., similarity) and are regarded as the same object. The constrained dilation results are shown in Fig. 10(c) in which some of the misjudged reliable depths have been patched using.

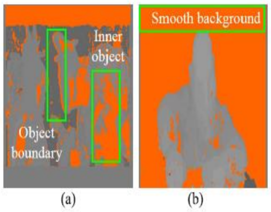


Fig. 11. Unreliable region classification. (a) Reliable map containing object boundaries and inner objects. (b) Reliable map including smooth background.

$$\begin{array}{c}\text{if}\ \ (R(x) == 0)\,\&\&\,(R(x-1) == 1)\,\&\&\\ ((I(x) - I(x-1)) < T_{\text{is}})\,\&\&\\ ((d_t(x) - d_t(x-1)) < T_{\text{ds}})\\ \text{then}\ \ R(x) = 1\end{array}$$

Where I (x) is the intensity value of a pixel with horizontal coordinate x, and Tis and Tds are the tolerance values for intensity and depth similarity, respectively. The proposed dilation method is not computationally expensive and is helpful for preserving object integrity. It also enhances the subsequent propagation phase. For example, the depth map shown in the upper-right corner of Fig. 10(b) shows the final depth results of the propagation phase without reliable depth dilation. In Fig. 10(b), the background depth is eroded into the foreground object, which resulted from a fragmentary object boundary, that is, if the object integrity is not fixed, the propagation phase may propagate the background depth into foreground objects. The dilation method preserves the object integrity effectively, and thus reduces the possibility of depth penetration from background to foreground objects, as shown in Fig. 10(c).

3.4. Efficient Reliable Depth Propagation

A bidirectional depth propagation scheme is developed to estimate the proper depth for unreliable regions. The proposed propagation scheme consists of three schemes: 1) horizontal propagation is introduced considering the foreground fattening effect [4], which is a side effect in window-based stereo matching; 2) vertical propagation using intensity similarity is employed to resolve interscanline inconsistency that appears in most raster-scan systems; and 3) background depth detection is applied for the case in which there is no reliable information in the whole scanline. To select the correct depth for unreliable segments, the unreliable segments are analyzed and divided into three categories (object boundary, inner object, and smooth background) as shown in Fig. 11. An object boundary consists of segments that are located between two objects with different depth values. Inner object segments imply that the unreliable region is located inside an object. A smooth background is a whole scanline that is unreliable.

The foreground fattening effect appears at the object boundary segment, that is, the foreground object expands into the background object. Under such a circumstance, unreliable segments are located in the background; thus the background depth is propagated into unreliable regions for reducing the fattening effect in object boundary segments as shown in Fig. 12.

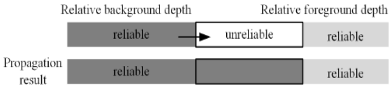


Fig. 12. Process of horizontal propagation.

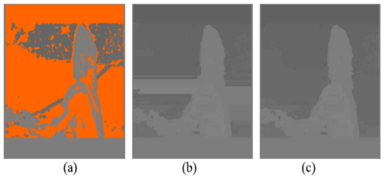


Fig. 13. Resolving interscanline inconsistency. (a) Reliable map and depth propagation considering (b) horizontal direction only and (c) both horizontal and vertical directions.

Interscanline inconsistency in raster-scan systems occurs after horizontal propagation, as shown in Fig. 13(b). Many scanline-based algorithms such as scanline optimization [4] and DP [20] cannot easily enforce interscanline consistency, thus produce horizontal streaks in the depth map. To handle this problem with the constraint of limited hardware storage resources, the proposed scheme checks the intensity similarity with the previous scanline to determine whether to apply vertical propagation. If the intensity of the unreliable segment is similar to that of previous scanline, the depth of the previous

scanline is propagated into the unreliable segment, as shown in Fig. 14. The vertical propagation is performed prior to the horizontal propagation. Assume that an unreliable segment stars at coordinate xus and ends at xue. The applied depth for propagation d (x) is selected as

$$\begin{array}{l}\text{if}\ \ \text{num}\big((I(i) - I'(i)) < T_{\text{is}}\big) > T_v//\text{vertical propagation}\\ \quad \text{for}\ x_{\text{us}} \le i \le x_{\text{ue}}\\ \qquad d(i) = d'(i)\\ \text{else}//\text{horizontal propagation}\\ \quad \text{for}\ x_{\text{us}} \le i \le x_{\text{ue}}\\ \qquad d(i) = \min(d_t(x_{\text{us}} - 1), d_t(x_{\text{ue}} + 1))\end{array}$$

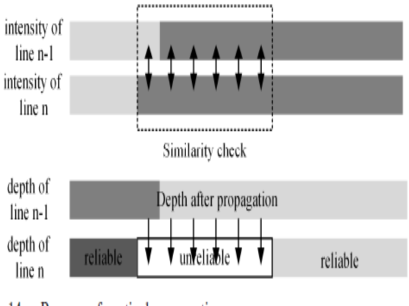


Fig. 14. Process of vertical propagation.

where I (x) and d(x) are, respectively, the intensity and depth values at the x coordinate, I _(x) and d_(x) are those in the previous scanlines, and Tv is the selected voting threshold close to (xue − xus). A smaller depth value in the aforementioned horizontal propagation indicates a farther scene. Fig. 13(c) shows the depth result obtained by applying both the horizontal and vertical propagation schemes.

For depth propagation in a smooth background region, because there is no reliable depth in the whole scanline, the depth is estimated by analyzing the reliable depth distribution of the image. In general, reliable depths of a smooth background appear sparsely at the boundary with a foreground object and the depth values are far from those of most foreground objects. Therefore, the developed background depth detector identifies a group of depths outside the main lobe of the depth distribution as the depth of a smooth background and then propagates them into the unreliable smooth background.

The physical meanings and properties of the parameters used in the developed algorithm are summarized in Table I.

4. HARDWARE ARCHITECTURE DESIGN AND IMPLEMENTATION

This section describes the proposed hardware architecture and optimization techniques. Without loss of generality, 7×7 window matching is used as an example. The same techniques can be directly applied to various window sizes. The system framework is a frame-buffer-less scanline-based architecture, that is, the hardware component can be directly inserted into the pixel-serial data path of the system without introducing frame buffer access and bandwidth competition. Each input image pixel is only accessed once in the continuous raster scan order. Therefore, a scanline buffer is added to store the data required for window-based computation.

4.1. Area-Efficient Memory-Merging Design

An area-efficient memory management scheme is introduced for reducing the overall storage cost. A general scanline memory area for different window sizes using both 55 nm and 0.11 μm processes are analyzed as shown in Fig. 16. The data width of 8 b is the conventional single scanline memory, but the data widths of 24, 56, and 104 b are the combined memory blocks by considering the memory merging for the window sizes of 3 × 3, 7 × 7, and 13 × 13, respectively. As can be observed from Fig. 16, the area reduction ratio adopting the memory-merging scheme is higher for the advanced process and large window sizes.

Fig. 17(a) shows a conventional memory allocation scheme for 7 × 7 window SAD computation. It is composed of seven individual scanline memory buffers and a 7×7 register array for SAD computation. Because the data address in each scanline memory buffer is directly mapped to the horizontal coordinate in the image, it is straightforward to get 7 pixels with the same address in the seven scanline memory buffers to the register array in a single cycle. After seven cycles, the data required for the first SAD computation is ready and the register update for the following calculation within the search range is performed in circular-in-place mode to minimize the register buffer usage.

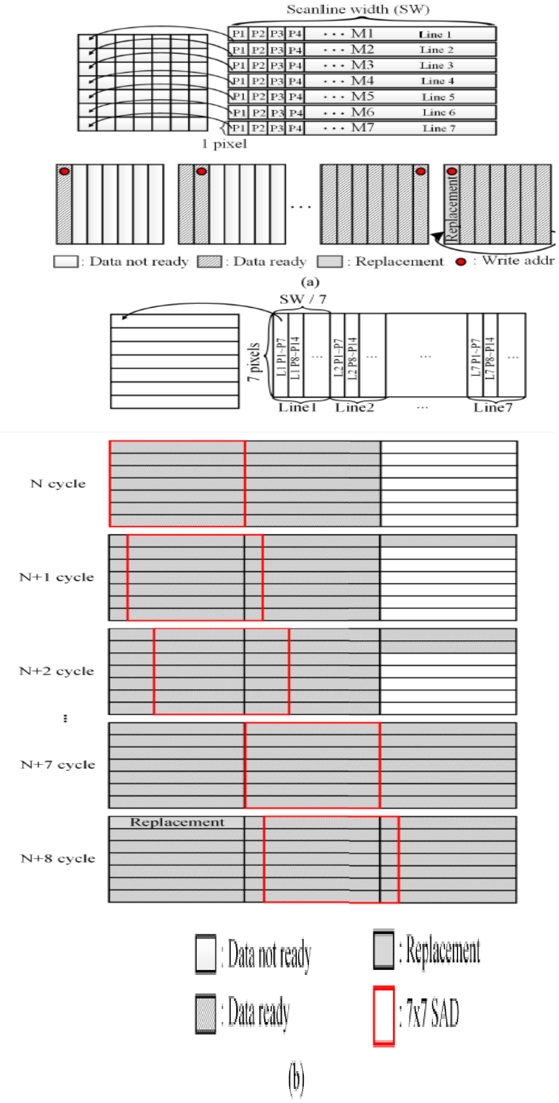

Scanline width (SW)
P1 P2 P3 P4 · · · M1 Line 1
P1 P2 P3 P4 · · · M2 Line 2
P1 P2 P3 P4 · · · M3 Line 3
P1 P2 P3 P4 · · · M4 Line 4
P1 P2 P3 P4 · · · M5 Line 5
P1 P2 P3 P4 · · · M6 Line 6
P1 P2 P3 P4 · · · M7 Line 7
1 pixel
Replacement
□: Data not ready ▨: Data ready ▒: Replacement ●: Write addr
(a)
SW / 7
7 pixels
L1 P1~P7
L1 P8~P14
L2 P1~P7
L2 P8~P14
L7 P1~P7
L7 P8~P14
Line1 Line2 ··· Line7
N cycle
N+1 cycle
N+2 cycle
N+7 cycle
N+8 cycle
Replacement
□: Data not ready
▒: Replacement
▨: Data ready
□: 7x7 SAD
(b)

Fig. 17. Scanline memory architecture. (a) Conventional seven distributed memory buffers implementation. (b) Merged single memory bank with data prefetching scheme.

Although conventional memory management is regular and simple, the area of seven individual scanline buffers is larger than that of a single large block because of the memory peripheral overhead. Therefore, for area efficiency, the memory width is increased from 1 to 7 bytes to merge the seven scaneline data into a large memory block, as shown in Fig. 17(b). In this configuration, the data obtained per cycle is 7 continuous pixels in a single scanline rather than 7 pixels in seven scanlines individually. To minimize the data access latency, three window buffers of size 7 × 7 are needed to prefetch the data while computing the matching cost with different disparity values. Using the proposed memory-merging technique, the total storage size for SAD computation can be reduced from a gate count of 39.1 k (37.1 k memory + 2 k window buffer) to that of 26.3 k (20.3 k + 6 k).

4.2. Resource Sharing for SAD Parallel Processing

To meet the real-time constraint, parallel SAD computation is required for a wide search range in high-resolution images. The resource sharing concept is used in the SAD computation to relax the rapid growth of complexity in parallel processing. Generally, adequate parallelism is determined according to the search range, the hardware specifications, and the real-time constraint. A proper disparity range for human stereopsis is 256 in Full HD resolution. For the most popular side-by side half 3-D format, the range is 128. Taking computation complexity into account, 2-by-2 image down sampling is tolerable for depth precision, and thus the relative disparity range becomes 64. Furthermore, considering the 7×7 matching window, the final search range is set to 56, which is a multiple of 7 and close to 64. To meet the requirement of 60 frames/s there are only ten cycles available for each pixel according to (12), assuming a clock frequency of 166 MHz and a resolution of 480 × 540

$$\text{Cycles per pixel} = \frac{\text{clock frequency}}{\text{resolution} \times \text{f/s}}.$$

As a result, 7-pixel parallel processing can meet the real-time requirement. The parallel SAD computation is shown in Fig. 18. Each PE computes the matching costs of one column in a 7 × 7 window by using

$$\text{PE}(m,n) = \sum_{n=1}^{W} |I_L(m,n) - I_R(m+d,n)|$$

Where I (m, n), d, and Ware, respectively, the intensity,disparity, and window size. The column indices of the left frame are represented as 1, 2, . .,13 (indexL) and those of the right frame are a, b, . . . ,m (indexR). The parallel processing set includes 7 pixels, denoted as {x, x +1, . . . , x +6}, and the used PEs are represented as PE(indexL, indexR). 49 PEs are needed for the 7-pixel parallel processing without PE sharing.

Because most PEs can be shared owing to the common computations of neighboring pixels as shown in Fig. 18, the number of PEs can be reduced from 49 to 13 for the parallel processing set.

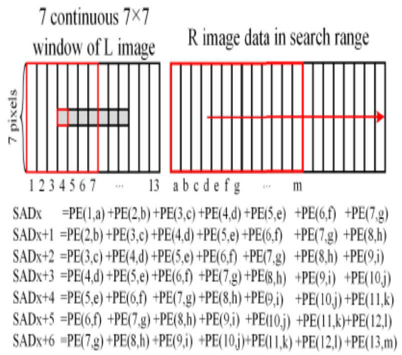

7 continuous 7×7 window of L image
R image data in search range
7 pixels
1 2 3 4 5 6 7 ··· 13
a b c d e f g ··· m
SADx =PE(1,a) +PE(2,b) +PE(3,c) +PE(4,d) +PE(5,e) +PE(6,f) +PE(7,g)
SADx+1 =PE(2,b) +PE(3,c) +PE(4,d) +PE(5,e) +PE(6,f) +PE(7,g) +PE(8,h)
SADx+2 =PE(3,c) +PE(4,d) +PE(5,e) +PE(6,f) +PE(7,g) +PE(8,h) +PE(9,i)
SADx+3 =PE(4,d) +PE(5,e) +PE(6,f) +PE(7,g) +PE(8,h) +PE(9,i) +PE(10,j)
SADx+4 =PE(5,e) +PE(6,f) +PE(7,g) +PE(8,h) +PE(9,i) +PE(10,j) +PE(11,k)
SADx+5 =PE(6,f) +PE(7,g) +PE(8,h) +PE(9,i) +PE(10,j) +PE(11,k)+PE(12,l)
SADx+6 =PE(7,g) +PE(8,h) +PE(9,i) +PE(10,j)+PE(11,k) +PE(12,l) +PE(13,m)

Fig. 18. Seven SAD computations in parallel, each for a dedicated pixel.

TABLE II

SYSTEM SPECIFICATION AND IMPLEMENTATION RESULTS

Items	Specification
Clock rate	166MHz
Image resolution	480×540
Disparity range	56
Frame rate	71 fps
Gate-count (excluding mem.)	183 k
Gate-count (including mem.)	223 k (6.72 kB memory)

Fig. 18. Seven SAD computations in parallel, each for a dedicated pixel.

TABLE II

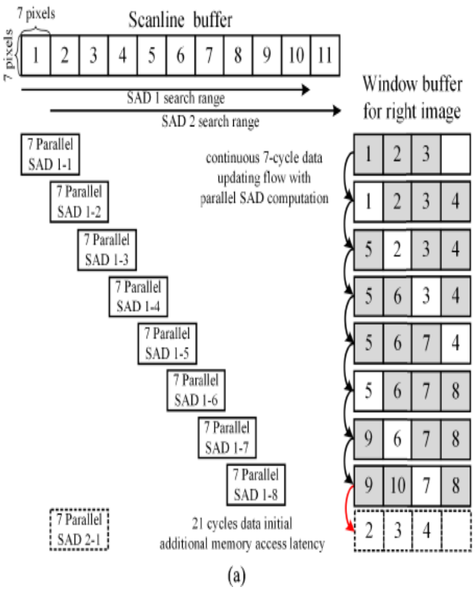
SYSTEM SPECIFICATION AND IMPLEMENTATION RESULTS

Items	Specification
Clock rate	166MHz
Image resolution	480×540
Disparity range	56
Frame rate	71 fps
Gate-count (excluding mem.)	183 k
Gate-count (including mem.)	223 k (6.72 kB memory)

4.3. Zig-Zag Memory Access for Latency Minimization

During the parallel SAD computation for a chosen parallel processing set, the data in the window buffer are updated simultaneously when the disparity estimation is searched from left to right or in reverse order. Because seven cycles are needed to update a 7×7 window register (as shown in Fig.17) and a continuous data flow is desired, one more 7×7 window register is added to maintain the seamless operations of parallel SAD computation for each parallel processing set. Therefore, it takes 21 cycles to initialize the window buffer, as shown in the right side of Fig. 19(a), assuming a left-to-right search sequence. In this figure, the notation SAD m–n stands for the nth step of the mth parallel processing set, and the numbers in the scanline buffer and window buffer are the indices of data blocks, each containing 7×7 pixels. To perform seven parallel SAD computations seamlessly for seven continuous disparity values, three data blocks (such as data blocks 1–3 initially) should be ready in the right window buffers. At the same time, new data (data block 4) are perfected to the idle (fourth) block of the window buffer for the next parallel SAD (SAD 1–2) computation. These steps are repeated until the overall search range is traversed in the window buffer.

When the current parallel processing set is switched to the next one (e.g., from SAD 1–8 to SAD 2–1), additional memory access latency may exist for updating the content of the window buffer. Specifically, if the search direction is fixed, 21 extra cycles are needed to initialize the data in the window buffer again when the parallel processing set changes, as shown in Fig. 19(a). This latency can be eliminated by adopting a zig-zag memory access scheme with data sharing. The search direction changes alternatively as shown in Fig. 19(b). The directions of zig-zag search are divided into in-order search (left to right) and reversed-order search (right to left). At the final step of the first set of parallel SAD computations (i.e., SAD 1–8), the idle block is updated with the new data (data block 11) required for the next set of parallel SAD computations (i.e., SAD 2–1). The zig-zag search order between two consecutive parallel processing sets allows the data in the window buffer at the final step of the current set (e.g., SAD 1–8) to be reused by the first two steps of the next set (accordingly SAD 2–1 and SAD 2–2). As a result, there is no additional latency penalty for switching between two consecutive sets of parallel SAD computations. The zig-zag memory access scheme minimizes latency and reduces the total processing time by approximately 25%.



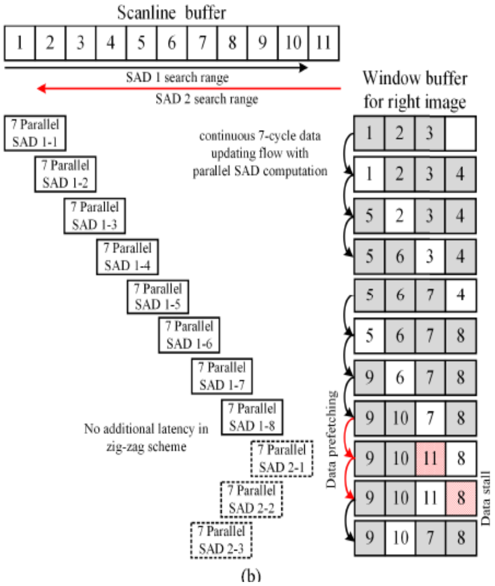


Fig. 19. Memory access latency minimization. (a) Schedule of conventional in-order unidirectional memory access. (b) Schedule of proposed zig-zag memory access.

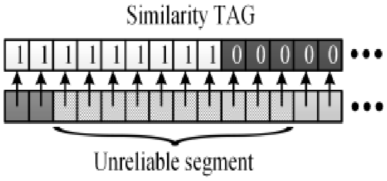


Fig. 20. Similarity tag used in depth propagation phase.

4.4. Preprocessed Key Indices and Region Similarity

The information required for depth reliability computation and depth propagation is obtained simultaneously while performing the SAD computation to avoid accessing image data repeatedly from the scanline buffer. As shown in Fig. 15, because the cost values within the search range are generated in a serial manner, only two simple comparators are used determined whether to update the minimum and the second minimum as well as the number of cost values in the predefined narrow band. The results are stored in registers for computing the Rinit and Rsmth. The similarity pre-computation unit is applied to obtain the intensity similarity with the previous line by checking the spatial difference in a 3 × 3 region. The required image data can be accessed directly from the window buffer while performing SAD computation. Finally, the similarity with the previous scanline is recorded using the binary tag as shown in Fig. 20.

In the proposed design, the SAD computation requests a large disparity search range and adopts a 7-pixel parallel architecture according to the constraint of real-time applications. On the contrary, the reliability computation has relatively low complexity that can be processed in a serial manner. Therefore, a 7-to-1 parallel-to-serial controller is required for passing the preprocessed information from the SAD computation unit to reliability computation unit. After acquiring the initial depth reliability decision Rinit using the minimum and the second minimum values, the deterministic unreliable depth is decided by considering: 1) Rsmth derived from the low-texture detection unit; 2) Rgocc determined using the histogram statistical results of the previous frame; and 3) Rlocc by checking the neighboring depth difference.

In the propagation phase, interscanline inconsistency can be improved using the similarity tag to determine whether to apply the depth from the previous scanline. A TAG with a value of 1 means that the depth of the previous scanline will be propagated downward to the unreliable segments in the current scanline. Finally, the remaining unreliable regions are filled with the nearest reliable depth at the horizontal direction. The computation of reliability decision and depth propagation is very simple with the preprocessed key indices. Moreover, the latency of sequentially processing 7 pixels is completely overlapped with the disparity search time of the SAD phase.

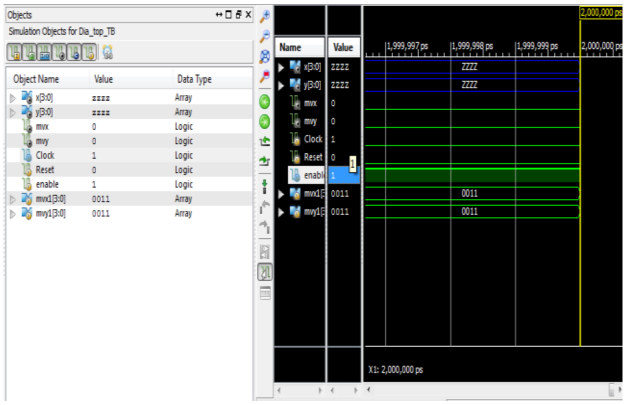
5. EXPERIMENTAL RESULTS AND PERFORMANCE COMPARISON

The proposed reliability-based stereo-matching architecture was coded in hardware description language and then synthesized using TSMC 90-nm CMOS technology with the typical library. Table II lists the core characteristics of the proposed design. The clock frequency can be up to 166 MHz, leading to a frame rate of 71 frames/s for a 480×540 resolution. The total gate count is 183 k excluding memory (6.72 kB) and is dominated by the SAD phase.

The hardware implementation results are compared with those from previous studies in Table III. Points times disparity per second (PDS), shown in (14) is a performance indicator used in the literature to evaluate the throughput and the disparity range at the same time. It is expressed as the product of the frame rate (f/s), the image resolution (Xres ×Yres), andthe disparity range (Drange)

$$PDS = Xres \times Yres \times f/s \times Drange.$$

As shown in Table III, the PDS value of the proposed algorithm is about 1030 million, which is much higher than those of previous designs. The area efficiency was computed by normalizing the PDS with respect to the gate count excluding memory size. The proposed reliability-based stereo matching algorithm performs well even when using only a small matching window, leading to a cost-effective hardware solution. The area efficiency is 11 times better than that in [7], whose algorithm was implemented with ADSW (adopting a31 × 31 window size).The results indicate that 39% of bad pixels in the trivial depth are removed with limited hardware overhead when using the proposed method. Moreover, the proposed algorithm and hardware architecture are window-size independent. Higher performance can be achieved if more hardware resources are available for handling larger window sizes. In addition to the evaluation using acknowledged patterns, a large number of test images from 3-D movie trailers which are the physical sequences of 3-D TV applications were also evaluated. Some results are shown in Fig. 21. Furthermore, all of the threshold values defined in Section III were fixed constants in the evaluation, which implies that the proposed algorithm satisfies most conditions without requiring fine tuning.



Objects
Simulation Objects for Dia_top_TB

Object Name	Value	Data Type
x[3:0]	zzzz	Array
y[3:0]	zzzz	Array
mvx	0	Logic
mvy	0	Logic
Clock	1	Logic
Reset	0	Logic
enable	1	Logic
mvx1[3:0]	0011	Array
mvy1[3:0]	0011	Array

Name	Value
x[3:0]	zzzz
y[3:0]	zzzz
mvx	0
mvy	0
Clock	1
Reset	0
enabl	1
mvx1[3	0011
mvy1[3	0011

Fig,21 proposed output

6. CONCLUSION

This paper has presented a cost-effective depth-reliability assisted algorithm and a scanline memory-merging architecture that can be applied to most of the window based stereo-matching techniques. The extracted reliability variables and bidirectional depth propagation improve most of the depth defects that are harmful to 3-D stereo applications, and reduce 39% bad pixels from using the 7 × 7 square window matching. The memory-merging architecture together with data prefetching schedule leads to a 32.7% reduction on the scanline buffer size. The design specification is suitable for frame-buffer-less systems with 60 frames/s FullHD side-by-side 3-D format (using 2-by-2 down sampling depth), a mainstream format for 3-D TVs. With the proposed schemes, introducing more aggressive stereo-matching algorithms to generate the initial depth map for further raising the depth quality is the possible extension of this paper.

REFERENCES

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms. New York, NY, USA: McGraw-Hill, 1990.

[2] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” in Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV), Jul. 2001, pp. 105–112.

[3] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 7, pp. 787–800, Jul. 2003.

[4] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense twoframe stereo correspondence algorithms,” Int. J. Comput. Vis., vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.

[5] O. Veksler, “Stereo correspondence with compact windows via minimum ratio cycle,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 12, pp. 1654–1660, Dec. 2002.

[6] R. K. Gupta and S. Y. Cho, “Real-time stereo matching using adaptive binary window,” in Proc. Int. Symp. 3D Data Process. Visualizat. Transmiss., vol. 1. 2010, pp. 1–8.

[7] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, “Algorithm and architecture of disparity estimation with mini-census adaptive support weight,” IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 6, pp. 792–805, Jun. 2010.

[8] K.-J. Yoon and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 650–656, Apr. 2006.

[9] Y. Qu, J. Jiang, X. Deng, and Y. Zheng, “Robust local stereo matching under varying radiometric conditions,” IET Comput. Vis., vol. 8, no. 4, pp. 263–276, Aug. 2014.

[10] L. De-Maeztu, A. Villanueva, and R. Cabeza, “Near real-time stereo matching using geodesic diffusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 2, pp. 410–416, Feb. 2012.

[11] M. Baydoun and M. A. Al-Alaoui, “Enhancing stereo matching with classification,” IEEE Access, vol. 2, pp. 485–499, May 2014.

[12] J. Joglekar, S. S. Gedam, and B. K. Mohan, “Image matching using SIFT features and relaxation labeling technique—A constraint initializing method for dense stereo matching,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 9, pp. 5643–5652, Sep. 2014.

[13] M. Hariyama, N. Yokoyama, and M. Kameyama, “1000 frame/sec stereo matching VLSI processor with adaptive window-size control,” in Proc. Conf. Asian Solid-State Circuits, Nov. 2006, pp. 123–126.

[14] M. Hariyama and M. Kameyama, “VLSI processor for reliable stereo matching based on window-parallel logic-in-memory architecture,” in IEEE Conf. Symp. VLSI Circuit Dig. Tech. Paper, Jun. 2004, pp. 166–169.

[15] S.-K. Han, S. Woo, M.-H. Jeong, and B.-J. You, “Improved-quality realtime stereo vision processor,” in Proc. 22nd Int. Conf. VLSI Design, Jan. 2009, pp. 287–292.

[16] A. Darabiha, W. J. MacLean, and J. Rose, “Reconfigurable hardware implementation of a phase-correlation stereoalgorithm,” Mach. Vis. Appl., vol. 17, no. 2, pp. 116–132, Apr. 2006.

[17] C. Ttofis and T. Theocharides, “Towards accurate hardware stereo correspondence: A real-time FPGA implementation of a segmentationbased adaptive support weight algorithm,” in Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE), Mar. 2012, pp. 703–708.

[18] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang, “Stereo matching with reliable disparity propagation,” in Proc. Int. Conf. 3DIMPVT, May 2011, pp. 132–139.

[19] N. Jacobson, Y. Freund, and T. Q. Nguyen, “An online learning approach to occlusion boundary detection,” IEEE Trans. Image Process., vol. 21, no. 1, pp. 252–261, Jan. 2012.

[20] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, “High-quality realtime stereo using adaptive cost aggregation and dynamic programming,” in Proc. Int. Symp. 3D Data Process., Visualizat., Transmiss., Jun. 2006, pp. 798–805.

[21] H. Hubert, B. Stabernack, and F. Zilly, “Architecture of a low latency image rectification engine for stereoscopic 3-D HDTV processing,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 5, pp. 813–822, May 2013.

[22] Middlebury Stereo Datasets accessed May 5, 2013.

[23] M. Werner, B. Stabernack, and C. Riechert, “Hardware implementation of a full HD real-time disparity estimation algorithm,” IEEE Trans. Consum. Electron., vol. 60, no. 1, pp. 66–73, Feb. 2014