

# FPGA Implementation of a Stereo Matching Processor Based on Window-Parallel-and-Pixel-Parallel Architecture

Masanori Hariyama, Naoto Yokoyama, Michitaka Kameyama  
Graduate School of Information Sciences  
Tohoku University

Aoba 6-6-05, Aramaki, Aoba, Sendai, Miyagi, 980-8579, Japan  
Email: {hariyama@,yokoyama@kameyama.,kameyama@}ecei.tohoku.ac.jp

Yasuhiro Kobayashi

Oyama National College of Technology  
771, Nakakuki, Oyama, Tochigi, 323-0806, Japan  
Email: y-kobayashi@oyama-ct.ac.jp

**Abstract**—This paper presents a processor architecture for high-speed and reliable stereo matching based on adaptive window-size control of SAD (Sum of Absolute Differences) computation. To reduce its computational complexity, SADs are computed using images divided into non-overlapping regions, and the matching result is iteratively refined by reducing a window size. Window-parallel-and-pixel-parallel architecture is also proposed to achieve to fully exploit the potential parallelism of the algorithm. The architecture also reduces the complexity of an interconnection network between memory and functional units based on the regularity of reference pixels. The stereo matching processor is implemented on an FPGA. Its performance is 80 times higher than that of a microprocessor(Pentium4@2GHz), and is enough to generate a 3-D depth image at the video rate of 33MHz.

## I. INTRODUCTION

Acquisition of reliable three-dimensional (3-D) images of a real scene plays an essential role in real-world intelligent systems such as intelligent robots and intelligent vehicles. Stereo vision is a well-known method to acquire 3-D information. The most important problem on stereo vision is to establish reliable correspondence between images. One commonly-used method to establish correspondence between images is the SAD(sum of absolute differences)-based method. The major problem on the SAD-based matching is that a window size for SAD computation must be large enough to avoid ambiguity but small enough to avoid the effects of projective distortions[1]. From this point of view, we have proposed the VLSI processor for stereo matching algorithm with variable window sizes[2],[3]. However, the processing time does not meet the performance requirement of the video rate of 33MHz. To meet the requirement, we present the new VLSI-oriented stereo matching algorithm that achieves high performance and high reliability. Firstly, to reduce the computational complexity of the variable-window-size algorithm, the stereo matching is executed using a reference image that is divided into non-overlapping regions. By using non-overlapping regions, the redundancy in computation is completely removed. Secondly, the window-parallel-and-pixel-parallel architecture is presented to fully exploit the parallelism. The major concern

is to design a simple interconnection network that supports parallel data transfer between memory modules and functional units. Since a scheduling provides the greatest impact on an interconnection network between and memory modules, a scheduling to exploit parallelism in a reference-and-candidate-window level is presented so that the same pixels are used even when the window size is changed.

The architecture is suitable not only for ASIC implementations but also for FPGA implementations where wiring delays are more dominant than logic delays in processing time. To demonstrate the efficiency, the stereo matching processor is implemented on an FPGA (APEX20KE, ALTERA Co.) for the images of the size  $64 \times 64$  and the maximum window size  $8 \times 8$ . The processing time is  $0.19msec$ . The performance is 80 times higher than that of a microprocessor(Pentium4@2GHz). Since the architecture has high scalability, it can be easily extended to applications that require larger image and window sizes.

## II. STEREO MATCHING ALGORITHM

### A. Basic SAD-Based Matching

Once correspondence between images is established, a 3-D point in the real scene can be found by triangulation. One commonly-used method for correspondence is a SAD-based one. Let us consider a reference window of a size  $W \times W$  centered at  $L(= (U_L, V_L))$  in the left image and a candidate window centered at  $(U_R, V_R)$  on the epipolar line in the right image as shown in Fig. 1. Then, an SAD in a window size  $W$  is given by

$$F_W = \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} |I_L(U_L + i, V_L + j) - I_R(U_R + i, V_R + j)| \quad (1)$$

where  $I_L$  and  $I_R$  are intensity values in the left and right images, respectively. If a candidate window exactly matches the reference window, then the SAD becomes 0. Given a reference pixel  $L$  in the left image, an SAD is computed for each candidate pixel on the epipolar line in the right image,

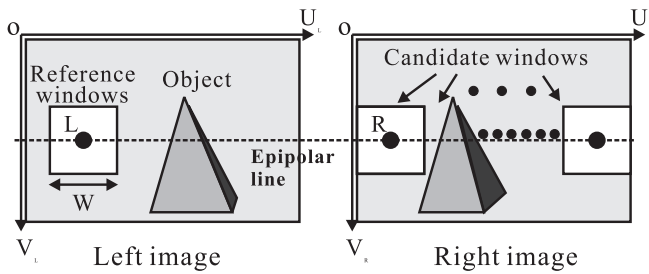


Fig. 1. Search for a corresponding point.

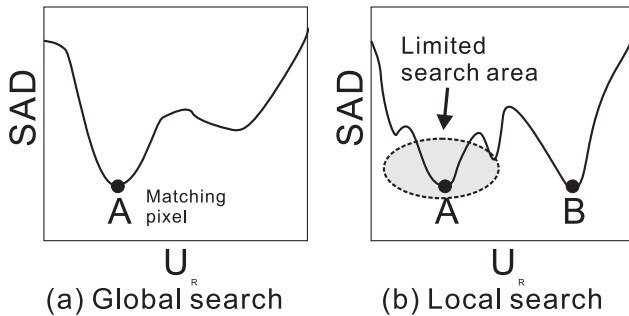


Fig. 2. SAD graphs.

and an SAD curve is obtained (Fig. 2). A pixel where the SAD curve becomes minimum is called a “matching” pixel.

The window size is an important parameter in the SAD-based method. If the window size is too small, there exist several possibilities for the choice of the corresponding pixel. On the other hand, if the window size is too large and the window includes pixels whose depths in the scene are different from each other, the matching pixel may not be the corresponding pixel due to different projective distortions in the left and the right images.

Moreover, computing an SAD requires  $W \times W$  absolute differences (ADs) and  $W \times W$  additions. Hence, straightforward SAD-based approach is time-consuming for a large window size in a straightforward manner.

### B. Reliable Stereo Matching with Variable Window Sizes

To solve the above-mentioned problem, we propose stereo matching algorithm that adaptively select a window size for each pixel. The algorithm is given as follows.

Step 1: Set the window size  $W$  to  $W_{max}$ , initially. The window size  $W_{max}$  is the maximum window size that is empirically determined such that the minimum peak of the SAD graph is unique. Divide the reference image into non-overlapping regions of the size  $W \times W$  as shown in Fig. 3. For each reference window  $RW$ , the corresponding window is searched for on the candidate image by computing SADs between  $RW$  and all the possible candidate windows. Note that the candidate windows are limited to the windows with the same vertical coordinate as the

reference window from the epi-polar geometry. As a result, the disparity is determined for each reference window, where the disparity is defined as difference between horizontal coordinates of the reference and the candidate windows ( $X_{RW} - X_{CW}$ ). The disparity is corresponding to the distance from the camera-coordinate-system origin.

Step 2: Shrink the window size to half, that is, set  $W \leftarrow W/2$  as shown in Fig. 4. Divide the reference image into non-overlapping regions of the size  $W \times W$ . The corresponding window of each reference window  $RW$  is found as follows.

Step 2-1 Let  $D_i$  ( $1 \leq i \leq 4$ ) be disparities of the regions, where the disparities are estimated at the larger window size around the reference window  $RW$  as shown in Fig. 4. For example,  $D_1$ ,  $D_2$ ,  $D_3$  and  $D_4$  are respectively  $D_{1,2}$ ,  $D_{1,3}$ ,  $D_{2,2}$  and  $D_{2,3}$ . It is likely that the true disparity of  $RW$  is one of  $D_i$  ( $1 \leq i \leq 4$ ) according to smoothness constraints [2].

Step 2-2 Compute SADs for only candidate windows that are within  $d$  from the location corresponding to  $D_i$  ( $1 \leq i \leq 4$ ) (Figure 4). The search area is limited by using the result at the larger window size to avoid the ambiguity at smaller window size. Determine the center pixel  $X_C$  of the candidate window with the minimum SAD value as the matching pixel of the center  $X_R$  of the reference window. By using the coordinates of the matching pixel, the disparity of  $P$  is computed by  $X_R - X_C$ .

Step 3: If  $W \neq 1$  then  $W \leftarrow W/2$ , and go back to Step 2. Otherwise, the latest matching pixel is determined to be the corresponding pixel.

Several experimental results show that the proposed algorithm provides the almost same quality, and reduces the computational amount to less than 0.5% compared to the variable-window-size algorithm without hierarchical approach [2].

### III. PROCESSOR ARCHITECTURE

There are two types of parallelism in stereo matching: window-level parallelism and pixel-level parallelism.

**Pixel-level parallelism:** ADs (absolute differences) in an SAD (Eq. (1)) is computed in parallel with pixels.

**Window-level parallelism:** SADs can be computed in parallel with reference and candidate windows.

From the view of the hardware implementation, the degree of the parallelism should be constant to achieve high-utilized ratio of hardware resources. In the proposed algorithm, the degree of the pixel-level parallelism is changed depending on a window size. For example, 16 AD units are necessary to compute an SAD for the window size  $4 \times 4$ , whereas only 4 AD units are sufficient to compute an SAD for the window

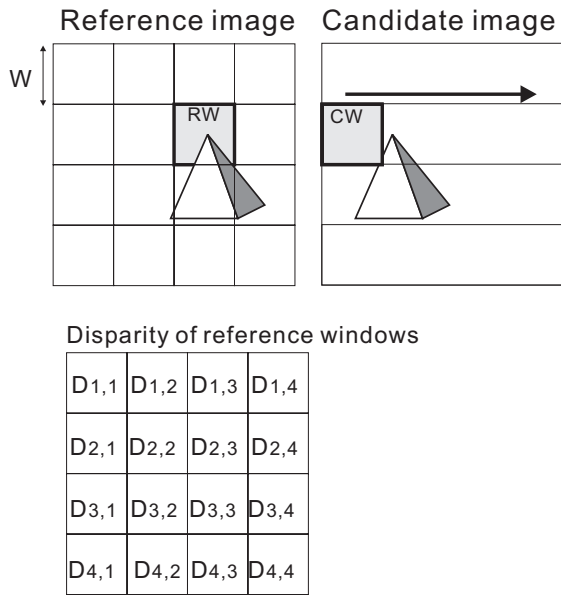


Fig. 3. Matching with a larger window size.

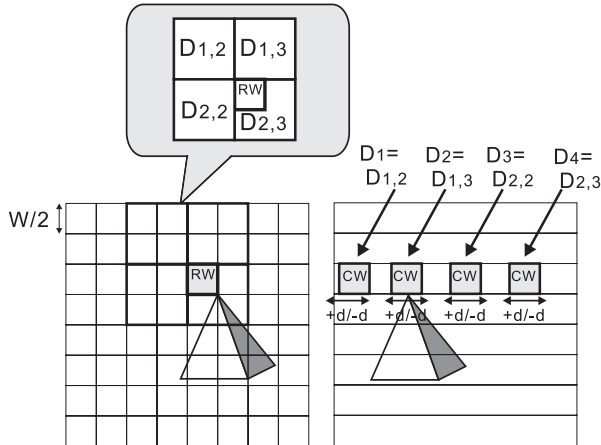


Fig. 4. Limiting the search area in the local search and matching with a smaller window size.

size  $2 \times 2$ . Therefore, 12 AD units are unused when only the pixel-level parallelism is used.

To solve this problem, the pixel-parallel parallelism is combined with window-parallel parallelism. Figure 5 shows the data-flow graph of the window-parallel-and-pixel-parallel (WPPP) scheduling for  $W_{max} = 4$ . A single SAD with 16 ADs is computed for the window size  $W = 4$ , whereas 4 SADs, each of which has 4 ADs, are computed for the window size  $W = 2$ . In our algorithm, the total number of ADs is kept constant although the window size is changed since the window size is iteratively shrink to half as shown in Figs. 3 and 4.

Figure 6 shows the overall architecture of the SAD unit. The SAD unit consists of memory modules, line buffers, processing elements. The candidate image is distributed among memory

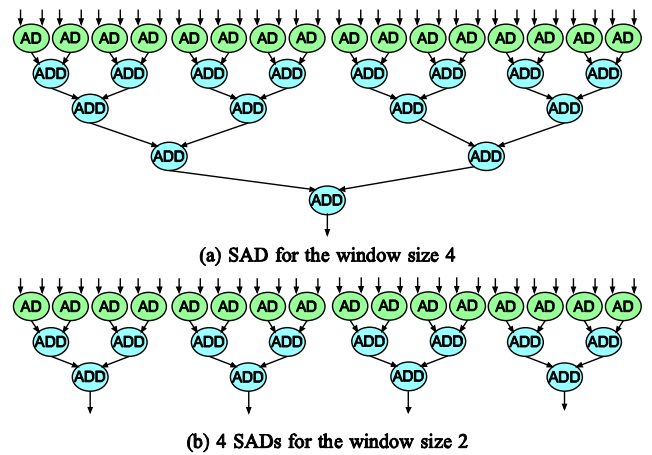


Fig. 5. Data-flow graph of the window-parallel and pixel-parallel scheduling.

modules C-MEMs, whereas the reference image R-MEMs. The number of C-MEMs and R-MEMs is  $W_{max}$ . The number of registers for a line buffer is same as the image size  $IW$ . The PEs are classified into 3 types: PE1, PE2, and PE3. The detailed structure of each PE is shown in Fig. 7. The PE1s are used to compute ADs. The PE2s and PE3s are used to add ADs and sum the results. Each PEs have a pipeline register so that AD computation and addition are overlapped in execution by PE-level pipelining. Each PEs have a search area controller and a minimum value detector to generate control signals locally. As a result, the delays for distributing control signals are significantly reduced. The major drawback of the proposed architecture is its large hardware amount. The number of PE1s are given by Eq.(2).

$$IW \times W_{max} \quad (2)$$

For example, the number of PE1s is 4096 for  $W_{max} = 16$  and  $IW = 256$ . Moreover, the number of I/O pins is large for parallel data transfer. To overcome these problems the SAD units is designed based on a bit-serial pipeline architecture. Bit-serial architecture is also useful for multi-chip implementation since it reduces the number of I/O pins between chips. The reason why the SAD unit has such a large number of PEs is balancing the performance and the area. Actually, the number of PEs can be reduced by reducing the degree of parallelism, namely, computing SADs sequentially. However, the total area of the SAD unit cannot be reduced since serial computation of SADs requires significant additional hardware such as register files to store intermediate results and image data although the execution time greatly increases. Let us explain about the utilized ratio of PEs. Computation of the SAD unit consists of following 3 steps.

- Step 1: Load pixel data of the reference region into registers.
- Step 2: Load pixel data of the candidate region into registers.
- Step 3: Compute SADs and detect the minimum value by shifting the pixel data of the candidate region.

In Step 1 and Step 2, PEs are in an idle state. In Step 3, the utilized ratio of PEs are ranging from 10% to 100% depending

TABLE I  
FEATURE OF THE STEREO MACHING VLSI PROCESSOR.

|                           |                               |
|---------------------------|-------------------------------|
| Image size                | 64 × 64 pixel(8bit-grayscale) |
| Max Window size           | 8 × 8 pixel                   |
| Device                    | APEX20KE (ALTERA Co.)         |
| Number of PEs(AD)         | 512                           |
| Frequency                 | 86 MHz                        |
| Processing time           | 0.19msec                      |
| Logic element consumption | 42570(82%)                    |

on the positions of PEs. As a result, the average utilized ratio of PEs becomes about 60%. Increasing the utilized ratio may be possible by reducing the number of PEs. However, it may result in a larger area because of an inter-PE network and registers to store intermediate results.

The proposed architecture is suitable not only for ASIC implementation but also FPGA implementation where the interconnection delays are more dominant than functional-unit delays since its interconnection network complexity is very low. To demonstrate the performance, we implement the stereo vision processor using an FPGA. Table I shows the features of the stereo vision processor. The FPGA APEX20KE(ALTERA Co.) and QuatousII(ALTERA Co.) are used as an FPGA device and a CAD, respectively. The image size and the maximum window size is slightly small for practical applications that require the image size of 256 × 256 and the maximum window size of 16 × 16. However, they can be extended by using multi-chip without decreasing performance due to the scalability of the proposed architecture. The maximum frequency and the processing time are 86MHz and 0.19msec, respectively. The processing time is smaller than 33msec, and it indicates that the 3-D image can be generated at the video rate. Moreover, the performance is 80 times higher than that of Pentium4@2GHz.

#### IV. CONCLUSION

The parallel stereo vision VLSI processor with a simple interconnection network is proposed. The key to success is the hierarchical approach using non-overlapping regions on the reference image, and the scheduling to exploit the parallelism in a reference-and-candidate-window level.

#### REFERENCES

- [1] S.T.Barnard, "Stereo vision, " in Encyclopedia of Artificial Intelligence. New York: John Wiley, pp.1083-1090(1987).
- [2] M. Hariyama, T. Takeuchi, M. Kameyama,"VLSI Processor for Reliable Stereo Matching Based on Adaptive Window-Size Selection" in Proc. International Conference on Robotics and Automation, pp.1168-1173(2001).
- [3] Masanori Hariyama and Michitaka Kameyama,"VLSI Processor for Reliable Stereo Matching Based on Window-Parallel Logic-in-Memory Architecture", Digest of Technical Paper 2004 Symposium on VLSI Circuits VLSI Symposium,pp.166-169(2004).

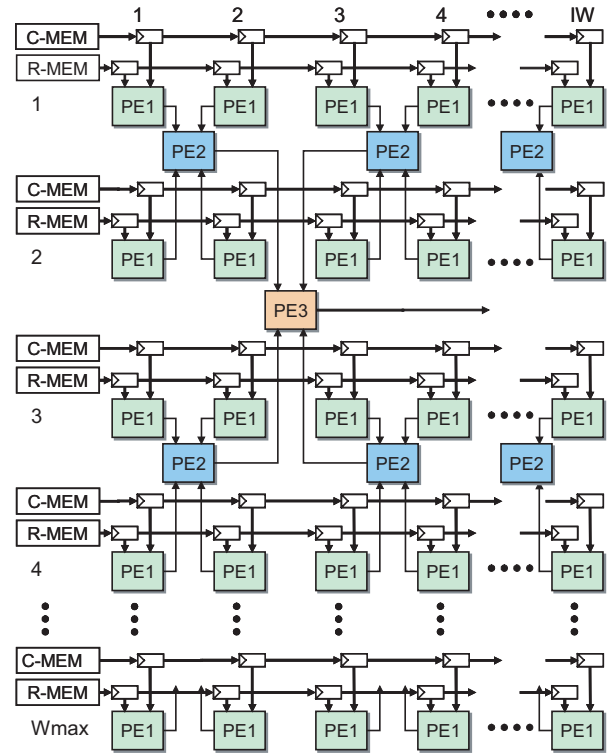


Fig. 6. Overall architecture of the SAD unit( $W_{max} = 4$ ).

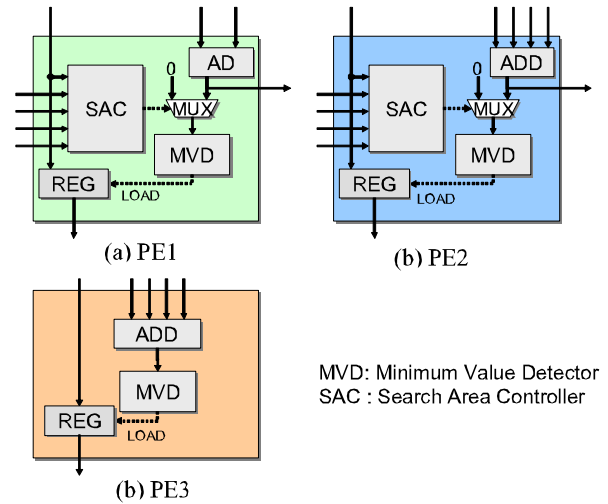


Fig. 7. Block diagram of PEs.