Interdisciplinary Information Sciences, Vol. 7, No. 2, pp. 289-297 (2001)

Pixel-Serial and Window-Parallel VLSI Processor for Stereo Matching Using a Variable Window Size

Masanori HARIYAMA and Michitaka KAMEYAMA

Graduate School of Information Sciences, Tohoku University
Aoba 05, Aramaki, Aoba, Sendai, 980-8579, Japan

Received March 23, 2001; Revised July 9, 2001

This paper presents a stereo-matching algorithm to establish reliable correspondence between images by selecting a desirable window size for SAD (Sum of Absolute Differences) computation. In SAD computation, a degree of parallelism between pixels in a window changes depending on its window size, while a degree of parallelism between windows is predetermined by the input-image size. Based on this consideration, a window-parallel and pixel-serial architecture is proposed to achieve 100% utilization of processing elements. Not only 100% utilization but also a simple interconnection network between memory modules and processing elements makes the VLSI processor much superior to conventional processors.

KEYWORDS: stereo vision, sum of absolute differences, ASIC, high-level synthesis

1 Introduction

To realize real-world intelligent systems such as intelligent robots and highly-safe vehicles, high-speed processing of a large mount of input data becomes important. However, the computational requirement exceeds computing power of present-day general-purpose processors. One promising way to overcome this problem is to develop special-purpose VLSI processors [1]-[4].

Acquisition of reliable three-dimensional (3-D) image of a real scene plays an essential role in real-world intelligent systems. Stereo vision is a well-known method of three-dimensional instrumentation. One important problem on stereo vision is to establish reliable correspondence between images. Another problem is that the correspondence search is time-consuming even if state-of-art general-purpose processors are used. To develop the highest performance VLSI processor, this paper presents a reliable stereo-matching algorithm and a new parallel architecture for its VLSI implementation.

In order to determine which pixel in one image (candidate image) matches a given pixel L in another image (reference image), we consider a rectangular window centered at L and compute a sum of absolute differences (SAD) for a candidate window of each possible location in the candidate image. If the reference window and the candidate window exactly match each other, then the SAD becomes 0. 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 [5]. To solve this problem, several algorithms have been reported until now [6]-[8]. However, these algorithms are not suitable for parallel processing since regularity and high-degrees of parallelism are not found in them. From this point of view, this paper presents a VLSI-oriented stereo matching algorithm with variable window sizes. The method is based on an idea that an SAD graph has a unique and clear minimum at the reliable matching pixel. A desirable window size that gives the reliable matching pixel is determined at each pixel based on the uniqueness of a minimum of an SAD graph. Moreover, the proposed algorithm has regular data flow based on iteration of SAD computation.

In designing a VLSI processor that executes the proposed algorithm, there are two major considerations One is to achieve high utilization of processing elements (PEs) for SAD computation. In SAD computation, a degree of parallelism between pixels in a window changes depending on its window size. Pixel-parallel SAD computation results in low utilization since many PEs may not be utilized for a small window size. To solve this problem, an SAD is computed in a pixel-serial manner where a single absolute difference (AD) is computed in each control step. The regular data flow of the pixel-serial computation makes it possible to fully utilize a PE for SAD computation. Moreover, in correspondence search, a degree of window-level parallelism is predetermined by an image width. Therefore, candidate windows of the equal number are assigned to each PE in advance so that PEs are fully utilized.

Another is to design a simple interconnection network with capability of efficient parallel communication. A memory allocation and a functional allocation are proposed to minimize complexity of interconnection network between memory modules and PEs under a condition of completely parallel data transfer. The processing time of the VLSI processor based on the pixel-serial and window-parallel architecture is estimated to be 60$m$sec for input images of a size 512 × 512. Its performance is more than ten thousand times higher than that of the general-purpose microprocessor (Pentium II 400 MHz).

2 Stereo Matching Algorithm

2.1 Basic Stereo Matching Algorithm

Figure 1 shows a camera geometry of a binocular stereo system. Two cameras with parallel optical axes are arranged along a straight line called a ''*baseline*''. Given a pixel $L(=(U_L, V_L))$ in the left image, let us find a 3-D point P which is projected onto L by perspective projection. It is mathematically guaranteed by imaging geometry that P is projected a pixel (U_R, V_R) on an ''*epipolar*'' line in the right image. Once a pixel (U_L, V_L) on the epipolar line in the right image is determined as the corresponding pixel, the 3-D coordinates of P can be computed from the 2-D coordinates U_L, U_R and $V_L(=V_R)$ by triangulation.

To establish the correspondence, a similarity measure must be computed which reflects how well the pixel L matches each pixel on the epipolar line in the right image. One commonly used similarity measure is a sum of absolute differences (SAD). 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 left 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, and an SAD curve is obtained as shown in Fig. 2. A pixel where the SAD curve has its minimum is called a ''*matching*'' pixel. In a straightforward method, a window size is empirically predetermined. The matching pixel in the window size is determined as the corresponding pixel.

The window size is an important parameter in SAD-based stereo matching. If the window size is too small, there exist several possibilities for the choice of the corresponding pixel. Therefore, the window size must be large enough to avoid the ambiguity. 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.

2.2 Reliable Stereo Matching with Variable Window Sizes

To overcome the above problem, a variable window size for each pixel in the image is used. In the method, as small a window as possible that will still produce the reliable matching pixel is used. As shown in Fig. 2, an SAD curve usually has multiple local minima. Let Q_1 be a matching pixel.h, and Q_2 be a pixel where the SAD curve has the second smallest value of all the local minima. Then, a reliability measure R of the matching pixel is given by

$$R = F_W(Q_1) - F_W(Q_2).$$

The main idea underlying the definition is that the matching pixel Q_1 is reliable if the SAD curve has a unique and

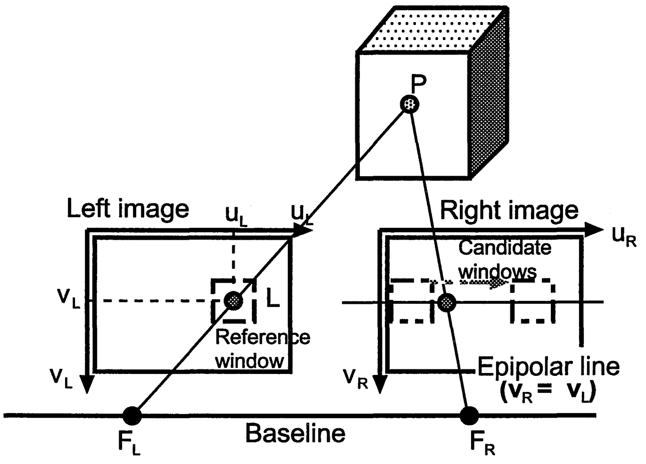


Fig. 1 Camera geometry for the stereo system.

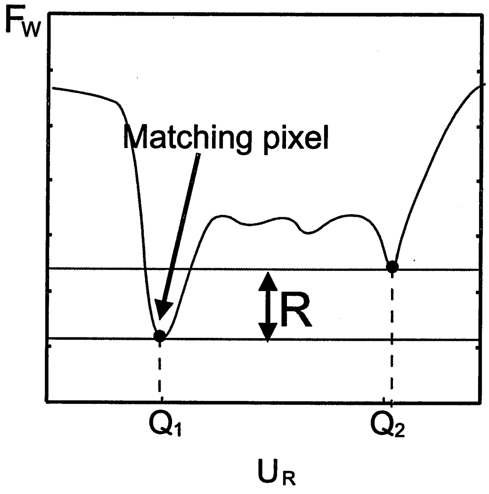


Fig. 2 SAD graph for a window size W.

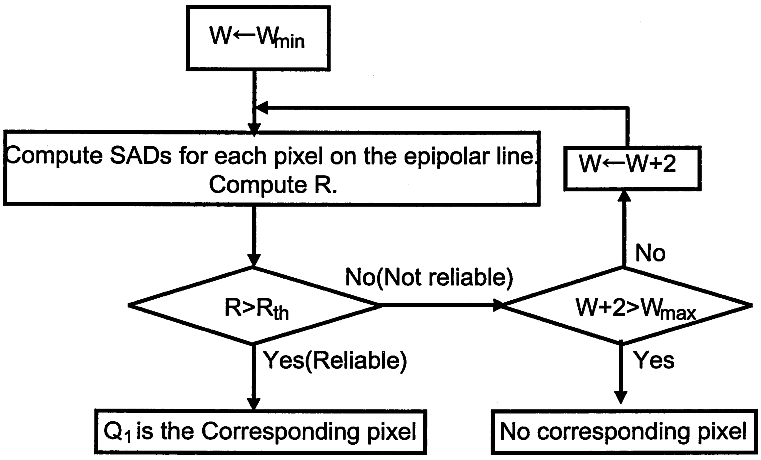
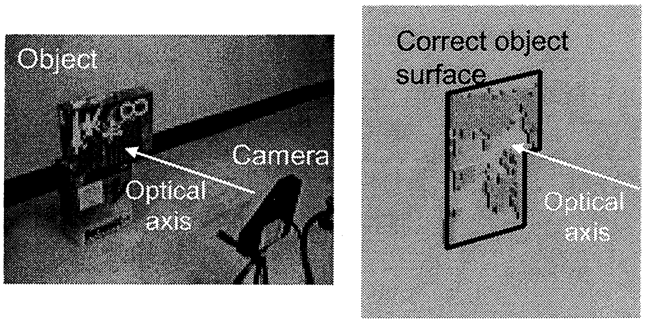


Fig. 3 Flowchart of the stereo matching algorithm.



(a)Experimental environment (b) Variable-window-size method(W=7 to W=25)

Fig. 4 Depth map of the front-parallel rectangular plane (Variable-window-size method).

clear minimum at Q_1. Figure 3 shows a flowchart for determining the corresponding pixel using the similarity measure. First, an SAD is computed for each pixel on an epipolar line for a window size W, and matching pixel Q_1 and the reliability measure R are obtained from the SAD curve (Fig. 2). Next, it is checked whether the matching pixel Q_1^W is reliable or not. Only if the R is large than the empirically-predetermined threshold R_{th}, Q_1 is reliable and is determined to be a corresponding pixel. Otherwise, the window is expanded, that is, W is set to $W + 2$. These steps are repeated until a corresponding pixel is found or the window size becomes larger than the maxi-

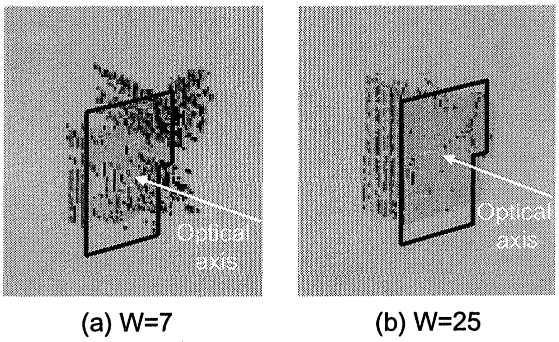


Fig. 5 Depth map of the front-parallel rectangular plane (Fixed-window-size method).

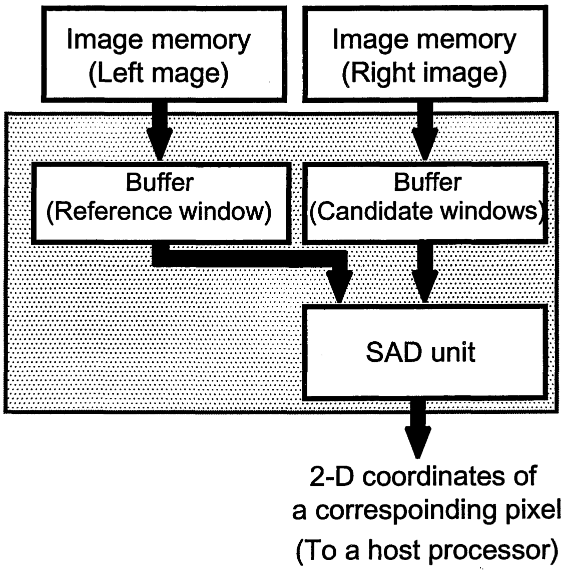


Image memory (Left mage)
Image memory (Right image)
Buffer (Reference window)
Buffer (Candidate windows)
SAD unit
2-D coordinates of a correspoinding pixel (To a host processor)

Fig. 6 Overview of a stereo vision VLSI processor.

mum window size W_{max}.

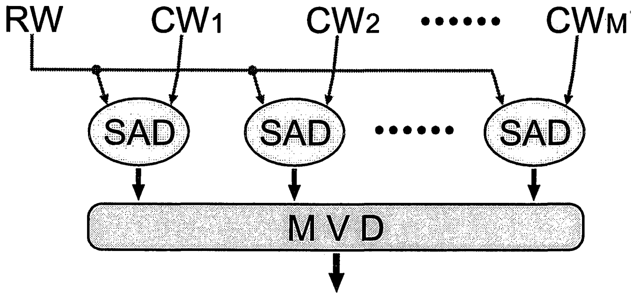
As an example of the stereo matching, a depth map of the front-parallel plane of a 3-D object (Fig. 4(a) is produced as shown in Fig. 4(b). Figures 5(a) and 5(b) show depth maps obtained by the fixed-window-size method for $W = 7$ and $W = 25$, respectively. The result clearly shows that the method selects reliable matching pixels.

3 Pixel-Serial and Window-Parallel Architecture

3.1 Overview

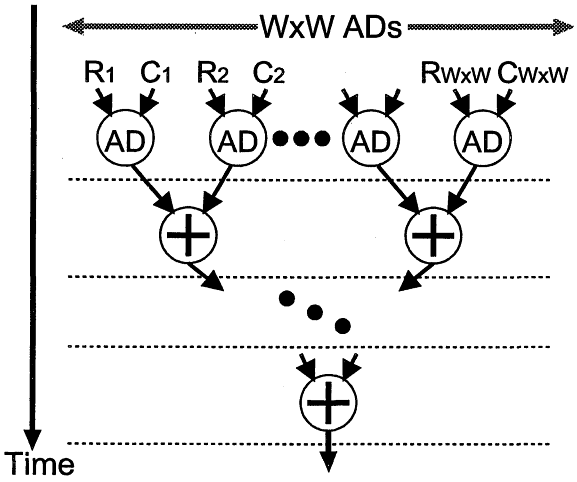
Figure 6 shows a block diagram of the stereo vision VLSI processor. It mainly consists of two image memories, buffers for a reference window and candidate windows, and a SAD unit. A Capacity of each image memory is too large to integrate them and the stereo matching unit on a single chip. In a typical case, each image memory has 256K-byte capacity for a 256-level gray-scale image with a size of 512 × 512. Therefore, images are stored in external memories. The external memories cause a data-transfer bottleneck due to its large access time. To solve this problem, frequently used pixels are stored in on-chip buffers with smaller access time as described in Section 3.3.

A corresponding pixel is searched as follows. Firstly, a reference window and candidate windows on an epipolar line are retrieved from image memories, and they are stored in buffers. Secondly, a corresponding pixel of a center pixel of the reference window is searched in the SAD unit. Finally, the resulting two-dimensional (2-D) coordinates of a corresponding pixel is send to a host processor that computes 3-D coordinates from the 2-D coordinates based on triangulation. The 3-D coordinate computation is executed by the host processor since its computational amount is small. The above mentioned steps are repeated for all the reference windows. All the steps are overlapped in execution by pipelining.



RW: Reference window CW: Candidate window
MVD: Minimum value detection

Fig. 7 Data-flow graph of stereo matching.



Ri: Pixel value of a reference window

Ci: Pixel values of a candidate window

W: Window size

Fig. 8 Data-flow graph of pixel-parallel SAD computation for a candidate window.

3.2 Stereo Matching Unit Based on a Pixel-Serial and Window-Parallel Scheduling

In the search for the corresponding pixel, there exist pixel-level parallelism and window-level parallelism.

Window-level parallelism. SADs can be computed in parallel for all the candidate windows on the epipolar line as shown in Fig. 7. The number M of candidate windows on the epipolar line is determined by the input image width, that is, M is fixed in advance. Therefore, it is relatively easy to exploit the window-level parallelism as described later.

Pixel-level parallelism. Absolute differences (ADs) in Eq. (1) can be computed in parallel for all the pixels in a candidate window. If an SAD is computed in a pixel-parallel manner as shown in Fig. 8, the number of ADs computed in parallel is changed depending on the window size W. This result in low utilization of circuits for AD computation. For example, let us compute SADs for a 9×9 window and an 11×11 window. In SADs for a 9×9 window and an 11×11 window, $81(= 9 \times 9)$ and $121(= 11 \times 11)$ ADs can be computed in parallel, respectively. Therefore, circuits for computing 40 ADs is not utilized during the computation of an SAD for a 9×9 window [10] when circuits for computing 121 ADs is used for pixel-parallel SAD computation.

To solve the problem, pixel-serial and window-parallel scheduling is proposed as shown in Fig. 9. An SAD is computed in a pixel-serial manner so that a single AD is computed in each step independently of the window size W. The drawback of the pixel-serial scheduling is that it requires more computational time than the pixel parallel scheduling. To reduce the larger computing time, the window-parallel scheduling is exploited so that SADs for different candidate windows are computed in parallel.

Figure 10 shows the SAD unit based on the pixel-serial and window-parallel scheduling. The SAD unit consists

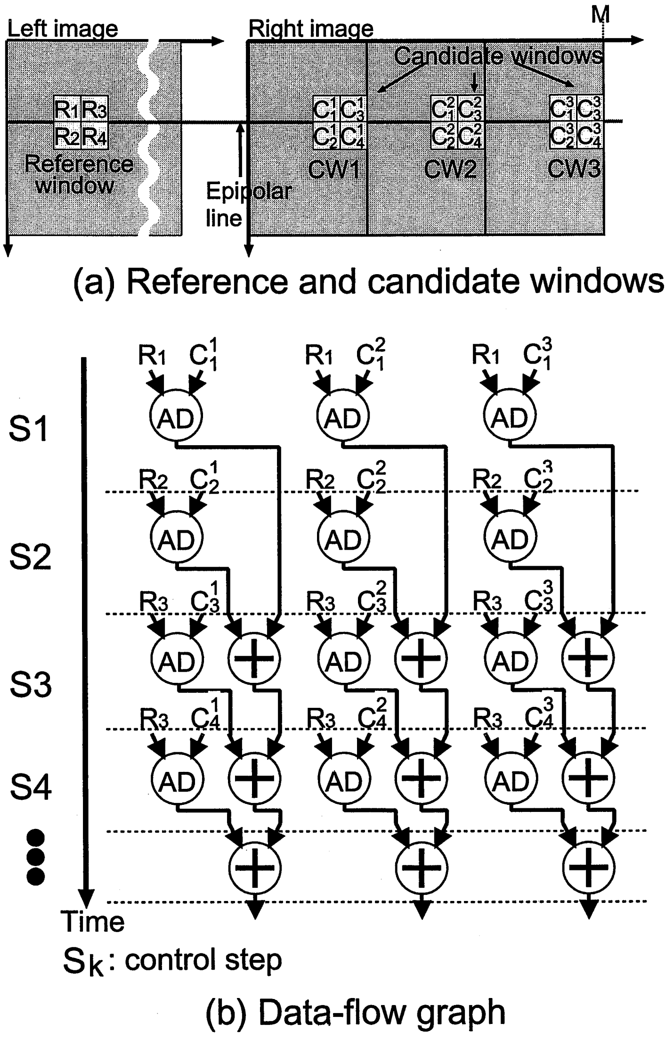


Fig. 9 Pixel-serial and window-parallel SAD computation.

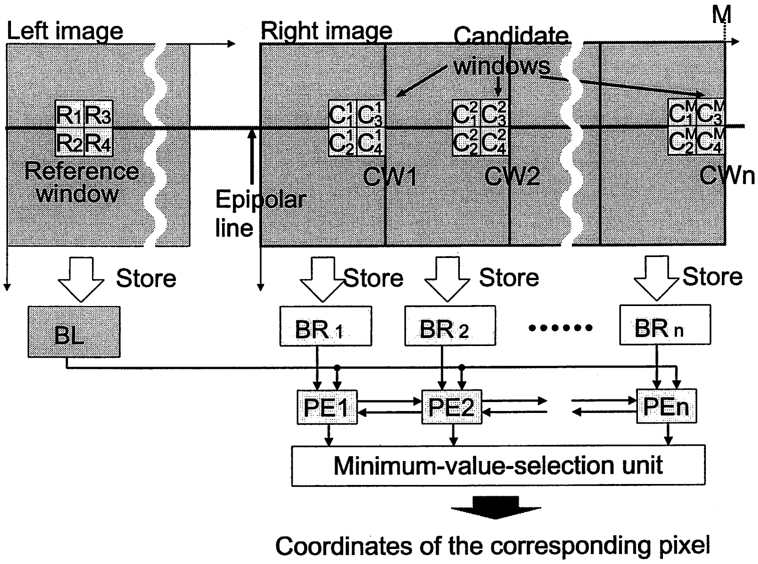


Fig. 10 Block diagram of the SAD unit for the window-parallel computation.

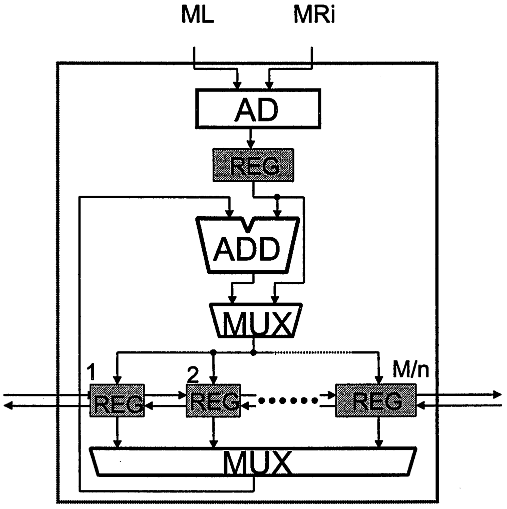


Fig. 11 PE for the pixel-serial SAD computation.

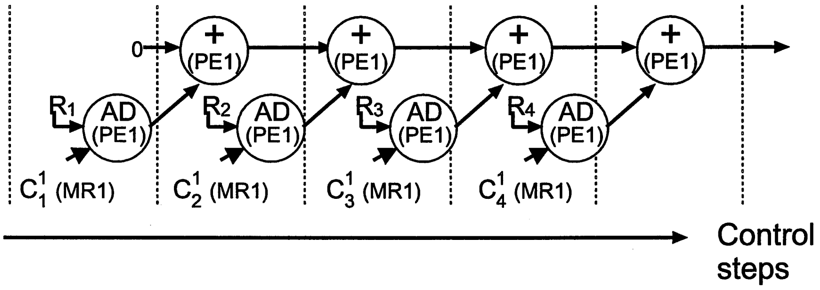


Fig. 12 Functional-unit allocation and a memory allocation for simple interconnection networks.

of identical n processing elements (PEs) and a minimum-value-selection unit. The n PEs computes SADs for n candidate windows in parallel in a pixel-serial manner. The minimum-value-determination unit computes the reliability measure R from the computed SADs and determines a corresponding pixel. Pixels in a reference window is stored in reference buffer BL. For parallel access, pixels in candidate windows on an epipolar line are equally distributed between memory modules $BR_i(i = 1, \cdots, n)$. Since the number M of candidate windows on the epipolar line is determined by the image width in advance, SADs for M/n candidate windows can be mapped in advance onto each PE so that all the PEs are 100% utilized. Figure 11 shows a block diagram of the PE with one AD circuit and one adder. The AD circuit and the adder in the PE can be utilized up to 100% since one AD and one addition are computed in each step based on the pixel-serial scheduling.

The major problem in designing the stereo matching unit is to find a simple interconnection network which support fast and efficient communication. Complexity of the interconnection network is determined by a scheduling, a memory allocation and a functional unit allocation [9]. To minimize the interconnection network between BL and PEs, ADs for the same reference pixel is computed in each control step as shown in Fig. 9(b). This scheduling results in a simple shared bus to transfer a single pixel in each control step from the M_L to all the PEs. Moreover, if all the ADs for a candidate window are computed in the same PE as shown in Fig. 12, there is no need to transfer intermediate results from one PE to other PEs. By this functional unit allocation, complexity of an interconnection network between PEs is minimized. Finally, to minimize the interconnection network between memory modules: $BR_k(k = 1, 2, \cdots,$ and $n)$ and a PE, all the candidate pixels used in a PE should be stored in a memory module. In other words, pixels in a candidate window should be stored in a memory module. To meet the requirement, M/n consecutive columns of the candidate image are stored in a memory module as shown in Fig. 10. By this memory allocation, a memory module BR_i is connected to only one PE.

3.3 Memory Architecture

Let us minimize a capacity of the candidate buffer. We assume that once a pixel P is stored in the candidate buffer, the candidate buffer keeps a pixel P until all the operations associated with P are finished. The assumption is introduced to minimize the number of memory accesses to the image memory.

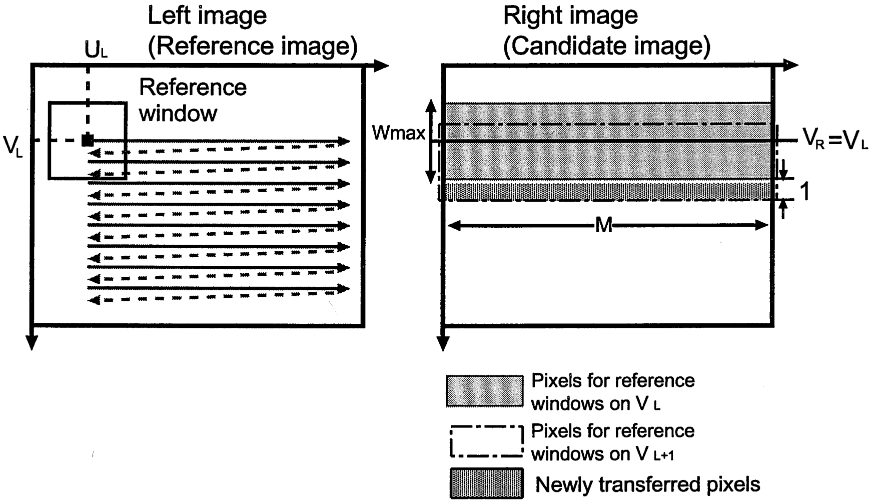


Fig. 13 Pixels stored in the candidate buffer.

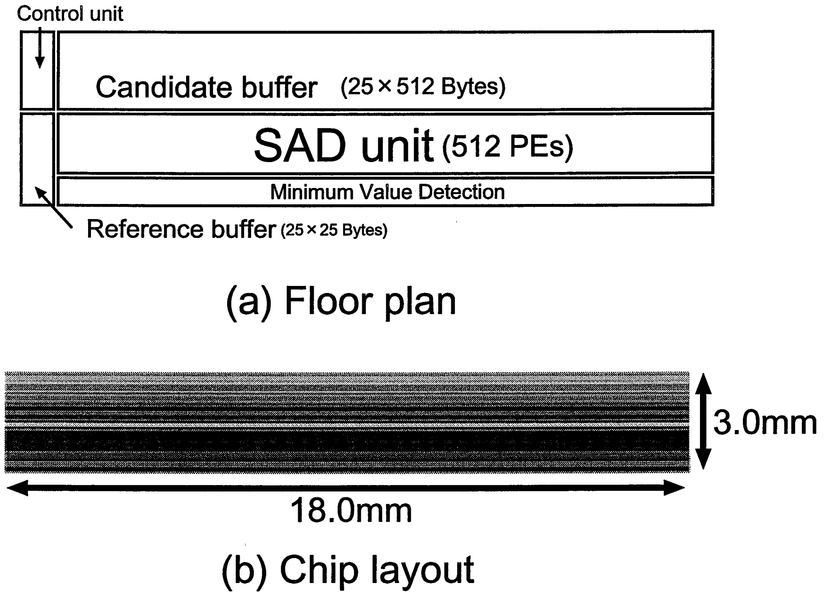


Fig. 14 Layout of the VLSI processor.

To minimize the buffer capacity, an ''*immediate output generation*'' scheduling is introduced. According to immediate output generation scheduling, once a pixel P is stored in the buffer, operations associated with P are executed as soon as possible. If all the operation associated with P are finished, there is no need to store P in the candidate buffer. That is, P can be replaced with a new pixel so that the buffer capacity is minimized.

As shown in Fig. 13, immediate output generation scheduling results in the raster-scan-oder execution. Let $RW(U_L, V_L)$ be a reference window centered at (U_L, V_L). Corresponding-pixel search (CPS)for $RW(U_L, V_L)(U_L = 1,2,\cdots, M)$ requires at most $W_{max} \times M$ pixels along an epipolar line $V_R = V_L$ (gray pixels in Fig. 13) so that these pixels stored in the buffer. Hence, CPS for $RW(U_L, V_L)(U_L = 1, 2,\cdots, M)$ is consecutively executed according to immediate output generation scheduling. After the CPS for $RW(U_L, V_L)(U_L = 1, 2,\cdots, M)$ is finished, CPS for $RW(U_L, V_L + 1)(U_L = 1, 2,\cdots, M)$ is consecutively performed to maximize the number of reused pixels. As a result, a reference window for CPS is selected according to a raster-scan oder. Moreover, only M pixels $(U_R, V_R + (W_{max} - 1)/2 + 1)(U_R = 1, 2,\cdots M)$ are newly transferred from the right image memory to the buffer as shown in Fig. 13.

Since overlap the transfer and CPS for $RW(U_L, V_L)(U_L = 1, 2,\cdots, M)$ by pipelining, the candidate buffer stores $(W_{max} + 1) \times M$ pixels.

Table 1. Features of the VLSI processor.

Technology	0.5-μm CMOS double-metal process
Area	18.0 × 3 mm^2
Input image	512 × 512 pixels (256-level gray scale)
Window size	From 3 to 25
Performance	60 *m*sec/depth map
Number of transistors	1 300 000
Clock frequency	200 MHz
Supply voltage	5 V

3.4 Evaluation

Figure 14 shows a floor plan and a chip layout of the stereo vision VLSI processor designed in a 0.5 μm CMOS process. It consists of an SAD unit, a candidate buffer, a reference buffer, a minimum value detection unit and a control unit. The candidate buffer has 512 × 26-byte capacity since 256-level gray-scale images are used as left and right images, and the maximum window size W_{max} is set to 25. This result shows that on-chip memory capacity of the right image can be reduced to 5.08% (= 26/512 × 100) based on immediate output generation scheduling in comparison with the case where the right image is stored in on-chip memory. A simple interconnection network between a memory module of the candidate buffer and a PE is achieved based on the memory and functional unit allocations so that interconnection delays are greatly reduced. Features of the VLSI processor are summarized in Table 1. The time required to produce a depth map estimated to be 60*m*sec for input images of a size 512 × 512. The performance of the VLSI processor is more than ten thousand times faster than the general-purpose processor (Pentium II 400 MHz).

4 Conclusion

Based on the window-parallel and pixel-serial architecture, not only a window size but also a window shape can be dynamically and flexibly changed. Therefore, the architecture can be applied to other VLSI processors for image processings. One useful application is a hierarchical image processing that uses images having different resolutions. For example, in object recognition, objects may be easily recognized in a low-resolution image since confusing detail features in the original image does not appear in the low-resolution image. This leads to a hierarchical approach where search for objects is begun at a low resolution, and refined at higher resolutions.

The window-parallel and pixel-serial architecture allows a simple interconnection network between on-chip image sensor so that totally bottleneck-free architecture will be achieved.

REFERENCES

[1] I. Masaki, S. Decker, A. Gupta, B. K. P. Horn, H-S. Lee, ''Cost-Effective Vision Systems for Intelligent Vehicles,'' in Proc. Intelligent Vehicles Symposium, pp. 39–43, 1994.

[2] P. G. Tzionas, A. Thanailakis and P. G. Tsalides, ''Collision-Free Path Planning for a Diamond-Shaped Robot Using Two-Dimensional Cellular Automata'', IEEE Trans. Robot. Automat., vol. 13, no. 2, 1997.

[3] M. Hariyama and M. Kameyama, ''Design of a Collision Detection VLSI Processor Based on Minimization of Area-Time Products'', in Proc. IEEE International Conference on Robotics and Automation, pp. 3691–3696, 1998.

[4] M. Ishikawa, T. Komuro, K. Ogawa, Y. Nakabo, A. Namiki, and I. Ishii, ''Vision Chip with General Purpose Processing Elements and Its Application,'' Int. Symp. on Future of Intellectual Intefrated Electronics, pp. 169–174, (1999).

[5] S. T. Barnard and M. A. Fischler, ''Stereo vision,'' in Encyclopedia of Artificial Intel-ligence. New York: John Wiley, pp. 1083–1090, 1987.

[6] T. Kanade and M. Okutomi, ''A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment,'' IEEE Trans. PAMI, vol. 16, no. 9, pp. 920–932, 1989.

[7] G. Sudhir, S. Banerjee, K. K. Biswas and R. Bahl, ''A cooperative integration of stereopsis and optic now computation,'' in Proc. ICPR, pp. 356–360, 1994.

[8] D. Scharstein and Richard Szeliski, ''Stereo Matching with Non-Linear Diffusion,'' in Proc. CVPR, pp. 343–350, 1996.

[9] D. Gajski, Nikil Dutt, Allen Wu, Steve Lin, ''High-Level Synthesis-Introduction to Chip and System Design,'' Kluwer Academic Publishers, pp. 259–296, 1992.

[10] S. Lee, M. Hariyama, M. Kameyama,'' A Three-Dimensional Instrumentation VLSI Processor Based on a Concurrent Memory-Access Scheme,'' IEICE Trans. Electron, vol. E80-C, No. 11, pp. 1491–1498, 1997.

[11] M. Okutomi, T. Kanade, ''A Multiple-Baseline Stereo,'' IEEE Trans. PAMI, vol. 15, no. 4, 1993.