실시간 Dense Disparity Map 추출을 위한 고성능 가속기 구조 설계

김 정 길[†] · Vason P. Srini[††] · 김 신 덕[†††]

요 약

본 논문에서는 위상기반 양안스테레오정합 알고리즘을 이용, 실시간으로 dense disparity map을 추출 가능한 고성능 가속기 구조를 설계하였다. 채택된 알고리즘은 웨이블릿 기반의 위상차 기법의 강건성과 위상상관 기법의 기본적인 control 기법을 결합한 Local Weighted Phase Correlation(LWPC) 스테레오정합 알고리즘으로서 4개의 주요 단계로 구성이 되어 있다. 해당 알고리즘의 효율적인 병렬 하드웨어의 설계를 위하여, 제안된 가속기는 각 단계의 기능블록은 SIMD(Single Instruction Multiple Data Stream) 모드로 동작하게 되며, 전체적으로 각 기능 블록은 파이프라인(pipeline) 모드로 실행된다. 그 결과 제안된 구조에서 제시된 파이프라인 동작 모드의 선형 배열 프로세서는 행렬 순차수행 방법에 의한 2차원 영상처리에서 전치메모리의 필요를 제거하면서도 연산의 일반성과 고효율을 유지하게 한다. 제안된 하드웨어 구조는 Xilinx HDL을 이용하여 필요한 하드웨어 자원을 look up table, flip flop, slice, memory의 소모량으로 표현하였으며, 그 결과 실시간 처리 성능의 단일 칩 구현 가능성을 보여주었다.

키워드 : 위상기반 스테레오 정합, 조밀 디스패러티 맵, 코프로세서 아키텍처, 심드 병렬처리

High Performance Coprocessor Architecture for Real-Time Dense Disparity Map

Cheong-Ghil Kim[†] · Vason P. Srini[††] · Shin-Dug Kim[†††]

ABSTRACT

This paper proposes high performance coprocessor architecture for real time dense disparity computation based on a phase-based binocular stereo matching technique called local weighted phase-correlation (LWPC). The algorithm combines the robustness of wavelet based phase difference methods and the basic control strategy of phase correlation methods, which consists of 4 stages. For parallel and efficient hardware implementation, the proposed architecture employs SIMD (Single Instruction Multiple Data Stream) architecture for each functional stage and all stages work on pipelined mode. Such that the newly devised pipelined linear array processor is optimized for the case of row-column image processing eliminating the need for transposed memory while preserving generality and high throughput. The proposed architecture is implemented with Xilinx HDL tool and the required hardware resources are calculated in terms of look up tables, flip flops, slices, and the amount of memory. The result shows the possibility that the proposed architecture can be integrated into one chip while maintaining the processing speed at video rate.

Key Words : Stereo vision, dense disparity map, local weighted phase correlation, SIMD

1. Introduction

Disparity computation to acquire 3-D depth information from a scene has been used for high level computer vision tasks such as navigation (robotics, cars, and space) and shape acquisition (virtual reality and movies). For this purpose, a stereo vision system can be devised using two cameras located at two different positions, which may imitate the human visual system known as binocular stereopsis that allows the visual sense to give an immediate perception of depth on the basis of the difference in points of view of the two eyes. It exists in those animals with overlapping optical fields, acting as a range finder for objects within reach. In stereo vision system, the geometry associated with solving this problem is simplified by assuming that the two cameras are coplanar with aligned image coordinate systems. (Fig.

[†] 정 회 원 : 연세대학교 컴퓨터과학과 BK21 연구교수
[††] 정 회 원 : UC Berkeley and Executive Director of Data Flux Systems Inc., Berkeley, CA.
[†††] 정 회 원 : 연세대학교 컴퓨터과학과 교수

논문접수 : 2007년 6월 27일, 심사완료 : 2007년 9월 5일

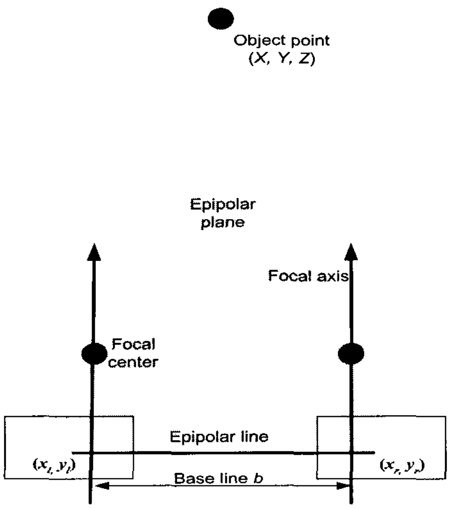
1) shows the basic structure for the stereo image formation and the stereo camera geometry. The center of the lens is called the camera focal center and the axis extending from the focal center is referred to as the focal axis. The line connecting the focal centers is called the baseline, *b*. The plane passing through an object point and the focal centers is the epipolar plane. The intersection of two image planes with an epipolar plane makes the epipolar line. Let *(X, Y, Z)* denote the real world coordinates of a point. The point is projected onto two corresponding points, (x_l, y_l) and (x_r, y_r), in the left and right images. The disparity is defined as the difference vector between two points in the stereo images, corresponding to the same point in an object, $v = (x_l - x_r$,; $y_l - y_r)$.

The most difficult area in stereo vision is matching points or features between the left and right images, called as stereo matching or stereo correspondence problem, which has been an intensive area of research for decades [1, 2]. Most of previous studies to solve and improve the performance of stereo matching can be grouped into three categories according to matching primitives; area based [3], feature based [4], and phased based approaches [5].

Area based approaches use the pixels or regions as the matching primitive to measure the similarity between two stereo images under assuming that the image intensity corresponding to a 3-D point remains the same in binocular images. That is, each location in the left image can find a similar location in the right image exploiting the epipolar constraint. This technique can produce dense disparity maps; on the other hands, it may have the disadvantage of becoming sensitive to contrast and illumination caused by using intensity values at each pixel directly.

Feature-based techniques use sparse primitives such as corners, edges, straight line segments, or other interesting operators. Here, the process can be divided into two stages; the first one involves preprocessing for extracting these features; the second one finds the corresponding from only these extracted features and assigns disparities to them. Therefore, this technique may extract more robust disparity map than area based approaches; however, it can not generate dense disparity maps.

In phase based techniques, the disparity is defined as the shift necessary to align the phase value of band-pass filtered versions of the two images. In [5], phase-based methods are shown to be robust when there are smooth lighting variations between stereo images. The first step in any phase-based method is to extract the phase from input images. One commonly used approach is to pass the input images through complex-valued quadrature pair filters. The phase of the complex-valued output of these filters is used as the primitive for stereo matching.



(Fig. 1) Basic structure for stereo image formation and stereo camera geometry

However, stereo matching, in general, requires considerably high computational expenses, especially when extracting dense disparity map which can produce accurate segmentation for more reliable applications. Therefore, the real time processing is still difficult on general purpose CPU or DSP; such that a specialized processing hardware is necessary to achieve the required computational complexity. Accordingly, this paper proposes high performance coprocessor architecture for real time dense disparity computation based on a phase-based binocular stereo matching technique called local weighted phase-correlation (LWPC) [6].

The rest of this paper is organized as follows. Section 2 reviews several researches of real time stereo implementation over the past decade. Section 3 introduces LWPC algorithm. Section 4 describes the architecture and operation of the proposed processor. Section 5 shows the simulation results. Finally, we conclude in Section 6.

2. Related Work

Even though stereo matching is a computationally expensive task especially when producing a dense disparity map, the fast advance of hardware technologies has enabled real time dense disparity map stereo to become possible. Until now most of researches for real time stereo matching have been achieved through special hardware implementations using either DSPs (digital signal processors) and FPGAs (Field Programmable Gate Arrays) or ASIC (application specific integrated circuit).

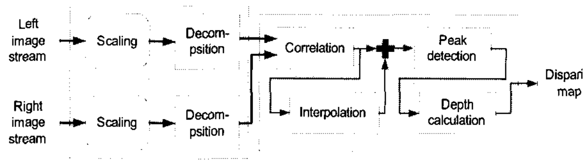
INRIA, a French national research institution, implemented a stereo system using normalized correlation method right -angle trinocular stereo configuration with the performance of processing 256 × 256 pixel images at approximately 3.6 fps [7]. Jet Propulsion Laboratory (JPL) developed a real time stereo system using a special image processing board on a 68040 CPU board [8]. This system was capable of processing approximately 1.7 fps with 256 × 240 pixel images. CMU (Carnegie Mellon University) developed the first prototype video-rate stereo machine which could process 30 fps with 256 × 240 pixel images on the custom hardware with an array of DSPs [9]. Interval Research Corporation in Palo Alto implemented the census stereo algorithm on the custom PARTS engine which consisted of 16 Xilinx 4025 FPGAs and 16 one megabyte SRAMs and computed 24 stereo disparities on 320 × 240 pixel images pixel images at 42 fps [10].

More recently, University of Toronto implemented a real time system using Transmogrifier-3A (TM 3A), a reconfigurable board containing four Xilinx Virtex2000E FPGAs [11]. Each FPGA is connected to the other three chips via a 98 bit bus. Each chip is also connected to a 256K x 64 bit synchronous SRAM memory, an I/O connector, and a bus which allows communication with a housekeeping FPGA. The system performs multi-resolution, multi-orientation depth extraction based on local weighted phase correlation. It can produce a dense disparity map of size 256 × 360 pixels with 8-bit sub-pixel accuracy disparity results at the rate of 30 frames per second.

In addition, RTSVP (real time stereo vision processing system), based on area correlation algorithms, was introduced with the implementation on FPGA [14] and the commercial graphic hardware was utilized for real time stereo vision processing [15].

3. Local Weighted Phase-Correlation Algorithm

The algorithm used to compute binocular disparity map in this paper is based on a phase-based stereo matching technique called local weighted phase-correlation (LWPC) [6]. It combines the robustness of wavelet based phase difference methods [5] and the basic control strategy of phase correlation methods [12]. This algorithm has the advantage of being implemented on the form of a dedicated hardware because it consists of simple computations of addition and multiplication and control flows. Therefore, once data is fed into processing pipeline, it goes through all of the computation stages without stopping until having final results, which enables real time data flow. (Fig. 2) shows the overall flow of LWPC algorithm which consists of four stages: scaling, decomposition, correlation, and peak detection.



(Fig. 2) Block diagram of LWPC algorithm.

In the scaling stage, the algorithm sub-samples input images by a factor of 2 horizontally and vertically at each level to build two lower scales of Gaussian pyramid. For this purpose, both left and right images are passed through the first quadrature-pair filters [13]. After that, all the scales are passed through multiple G2-H2 quadrature filter pairs; each of them has unique directions using steerable filters in the orientation decomposition stage [13]. That is, three quadrature pair filters are applied at each level tuned to orientations 0, +45°, and -45° where 0 is vertical. Assuming that $K_j(x)$ is the filter impulse response of the jth orientation on each pixel position x in the image, we can write the complex-valued output of the convolution with each scale of left and right images, $I_l(x)$ and, $I_r(x)$, as:

$$O_l(x)=K_j(x)\otimes I_l(x)\ ,\quad O_r(x)=K_j(x)\otimes I_r(x)\ . \qquad (1)$$

After that, the left and right pairs of filter outputs for each scale and each orientation are passed through the phase correlation block which assigns similarity measures or voting functions between a pixel in one image and its shifted versions in the other image. At this time, the voting function, $C_{(j,s)}(x,\tau)$, is defined as:

$$C_{(j,s)}(x,\tau)=\frac{W(x)\otimes[O_l(x)O_r*(x+\tau)]}{\sqrt{W(x)\otimes|O_l(x)|^2}\sqrt{W(x)\otimes|O_r(x)|^2}}\ , \qquad (2)$$

where $W(x)$ is a small localized window, τ is the pre-shift of the right filter output, and the subscript j refers to the j^{th} filter.

These voting functions are then combined over all the scales, $1\leq m\leq M$, and orientations, $1\leq j\leq F$, to build the overall voting value. Here F is the total number of orientations and M is the total number of scales. This can be expressed for the position, x, as below:

$$S(x,\tau)=\sum_{j,m}C_{(j,m)}(x,\tau) \tag{3}$$

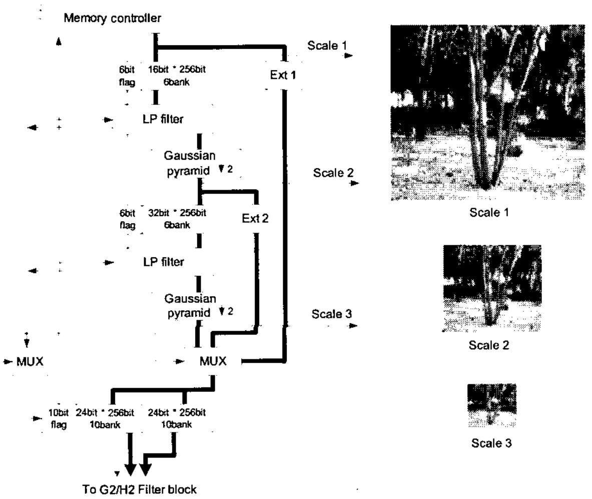
Finally, the shift corresponding to the location of the peak response will be selected as an estimate for the disparity.

4. Proposed Hardware Architecture

The proposed hardware architecture for extracting real time dense disparity map consists of several units according with those algorithm stages shown in (Fig. 2), scaling unit, orientation decomposition unit, phase-correlation unit, and interpolation and peak detection unit. Input data is gray scale raw image and the system generates 8-bit sub-pixel disparities on 256 by 360 pixel images. For parallel and efficient hardware implementation of the stereo depth computation, some modifications on the original LWPC are required such as fixed-point data representation and decreasing the low-pass filters.

4.1 Scaling unit

Scaling unit reads input data from frame buffer and down-samples the original image in two steps, each time by a factor of 2 in both horizontal and vertical directions. Such that the results are two Gaussian pyramids for left and right image, respectively. To avoid aliasing caused by a result of down-sampling, we pass the input image through a low-pass anti-aliasing filter. Here, a three-tap Gaussian FIR filter is used. (Fig. 3) shows the block diagram of scaling unit.



(Fig. 3) Block diagram of scaling unit

4.2 Orientation decomposition unit

In this unit, G2-H2 filter is used for orientation decomposition. G2-H2 filters are complex valued quadrature-pair filters and steerable, which means any arbitrary orientation of G2 or H2 filters can be expressed as a linear combination of a set of basis filters [13]. The basis set for G2 and H2 filter has three and four filters, respectively. In hardware, we have implemented all the seven basis filters using seven separable 7 × 7 FIR filters and then, by combining the basis filter outputs with proper coefficients, we construct two oriented filters in 45° and -45° degrees. Filter outputs are reduced to a 16-bit representation before being sent to the phase-correlation unit. (Fig. 4) shows the architecture of the orientation decomposition stage using seven filters.

The important advantage of using G2-H2 filters for hardware implementation is that they are separable, which results in less hardware resources than non-separable filters of the same size. A separable filter has an impulse response that can be expressed as the product of two functions: one which only depends on row, and one which only depends on column. Consider a separable filter with impulse response *K[x,y]*, which can be expressed as:

$$K[x,y]=F[x]\circ G[y] \tag{4}$$

Then, the 2-D convolution of image *I[x,y]* with *K[x,y]* can be written as:

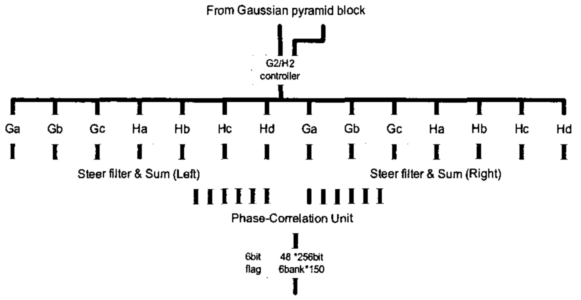
$$I[x,y]\otimes K[x,y]=(I[x,y]\otimes F[x])\otimes G[y] \tag{5}$$

Or

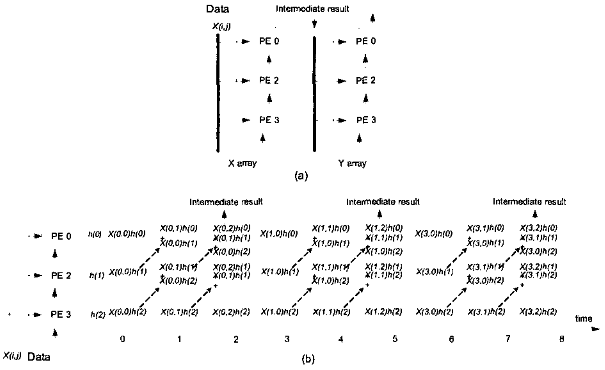
$$I[x,y]\otimes K[x,y]=(I[x,y]\otimes G[y])\otimes F[x] \tag{6}$$

Therefore, the convolution of the input image with an N x N, can be replaced with two separate 1-D convolutions with a horizontal vector, 1 x N, and a vertical vector, N x 1. This feature reduces the filter complexity from $O(N^2)$ to $O(2N)$.

However, in hardware, these approaches need to transpose the intermediate results using a shared memory array, which leads to a high circuit complexity and a long time for loading and unloading. This paper proposes a *pipelined array* architecture consisting of two linear arrays (X and Y) as shown on (Fig. 5(a)). (Fig. 5(b)) shows the overall time schedule of the proposed array. The intermediate result from X array can be immediately fed into Y array without additional control. Assuming that the X array starts computation at t = 0, the Y array



(Fig. 4) Block diagram of orientation decomposition unit

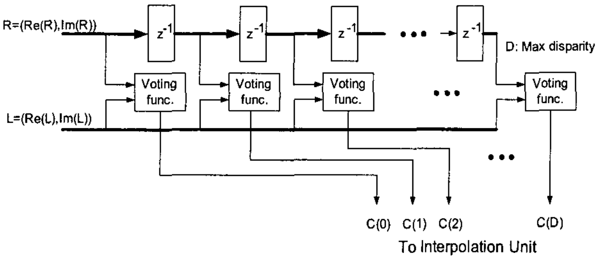


(Fig. 5) (a) Proposed pipelined array (b) Time schedule table

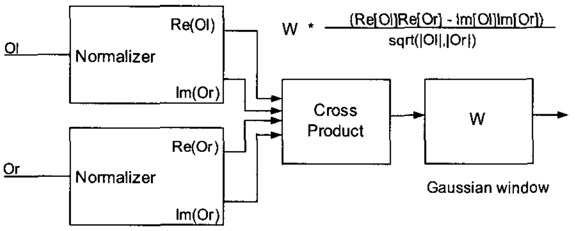
can start processing after the first intermediate result from X array. After that two arrays can operate at the exact same rate and generate results simultaneously. Performance gain can be obtained by the ratio of the total number of computations to the product of latency times and number of processors. Each PE (processing element) in the array consists of fixed point numerical arithmetic units, a shifter, a register file, and special registers for communications. For MAC (multiply and accumulate) operation, the result of multiplier is bypassed to adder in arithmetic units.

4.3 Phase correlation unit

(Fig. 6) shows the overall architecture of the phase correlation unit, in which the value of D represents the maximum disparity distance between left and right images. This unit computes the real part of the voting function using Equation 2 and finds the best match with a similarity function for each pixel in the left image and the horizontally shifted locations of that pixel in the right image. This is possible because for each pixel in one image, the corresponding pixel in the other image lies on the same scan line and within a maximum distance. The similarity function results are then combined across all scales and orientations. The shift value which produces the highest similarity will be selected as the best match.



(Fig. 6) Block diagram of phase correlation unit

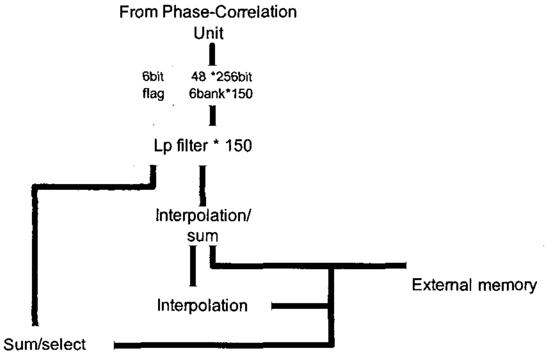


(Fig. 7) Block diagram of phase correlation unit

From Equation 2, to compute voting function C in location x of the image and for candidate disparity of τ, we need Gaussian windows, *W(x)*, at three different locations. However, for efficient hardware implementation, they are reduced to one by moving the window at the end of divider as shown in (Fig. 7).

4.4 Interpolation and peak detection unit

The interpolation and peak detection unit interpolates two coarser scale voting functions, $C_{(j,2)}(x,\tau)$ and $C_{(j,3)}(x,\tau)$, in both x and τ domain such that they can be combined with the finest scale voting function $C_{(j,1)}(x,\tau)$. (Fig. 8) shows the block diagram of the interpolation and peak detection unit. Here, the interpolated voting functions are then added together to produce the overall voting function $S(x,\tau)$. Such that the peak detection result of each pixel x in the image can be found with the value of τ when $S(x,\tau)$ is maximum. After that a simple

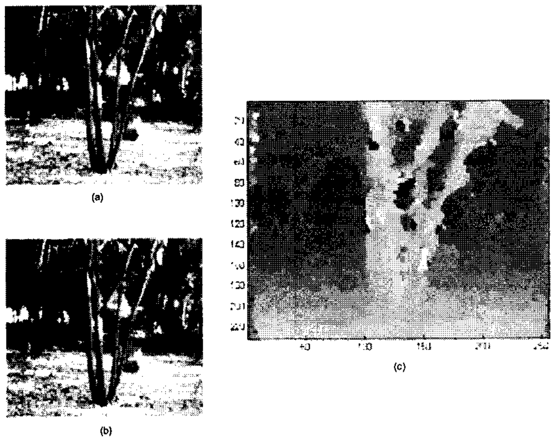


(Fig. 8) Block diagram of interpolation and peak detection unit sub-pixel peak detection scheme [11] refers to the maximum value, in which two adjacent points - left and right - in the τ domain are added to the maximum value and fitted to a quadratic curve. This result produces disparity values with 8-bit resolution - 5 bits for the integer and 3 bits for sub-pixel - from 20-pixel disparity range.

5. Experimental Results

We implemented the original LWPC algorithm using Matlab 7.0 to evaluate its functionality; and then simplified and optimized it for hardware implementation and emulated hardware functional behavior in software using C++. After that, we built the hardware based on the emulation version with VHDL [16] and synthesized using Xilinx ISETM 8.1i [17]. The execution times of two software platforms are compared on 1.7 GHz Pentium IV personal computer. And the hardware resources used in each unit of the proposed stereo system are calculated in terms of look up tables (LUTs), flip flops, slices, and the amount of memory.

First, to evaluate the functionality of the LWPC algorithm, we implemented the algorithm with floating point operations using C++ and Matlab 7.0. Their execution times are 35 and 22 seconds for a frame with Matlab and C++ implementation. In case of C++ implementation, we also measure the execution time after modifying the algorithm with fixed point representation and reducing Gaussian windows as mentioned in Section 4.3. These optimizations result in the further improvement on execution time about 7 seconds. However, it is still far from the real time processing on general purpose computing devices. The calculated sample disparity map using the simplified LWPC is shown on (Fig. 9(c)) with an input image pair of SRI tree, in which the distance is coded by grey scale and the color of closer objects become bright.



(Fig. 9) (a) Left input image, (b) Right input image, (c) Disparity map

<Table 1> Hardware resources with floating point operations

Unit		# of 4-input LUTs	# of flip-flops	# of slices	Multiplier (18*18)	Memory bank	External memory
Low pass filter adder	8-bit	676	432	555	N/A	10	N/A
	16-bit	1,320	848	1,089	N/A	5	N/A
G2/H2		32,780	4,064	19,643	266	54	N/A
Correlation	Filtering	54,275	31,800	43,575	N/A	375	N/A
	Exe.	5,907	16,986	4,884	75		N/A
Interpolati on	Inter 0	654	540	279	N/A	N/A	4M
	Inter 1	1,554	1,704	870	N/A	N/A	
Peak detection		1,580	1,002	795	N/A	N/A	N/A

<Table 2> Hardware resources with non-floating point operations

Unit		# of 4-input LUTs	# of flip-flops	# of slices	Multiplier (18*18)	Memory bank	External memory
Low pass filter adder	8-bit	350	60	203	N/A	10	N/A
	16-bit	590	100	352	N/A	5	N/A
G2/H2		16,851	4,064	12,125	60	54	N/A
Correlation		17,554	6,600	13,170	260	180	N/A
Interpola-tion	Inter 0	2,316	540	2,603	20	26	N/A
	Inter 1	4,915	1,704	4,406	38	25	
Peak detection		1,178	813	607	N/A	25	N/A

<Table 1> lists the hardware resources for the stereo systems with floating point operations in terms of LUTs, flip flops, slices, and the amount of memory. Here, the size of input image is 8-bit grey scale and the final result of scale orientation unit is represented with signed 16-bit values. The maximum displacement in the correlation unit is 20 pixels to find the best match. Furthermore, to reduce computation complexity, Equation 2 is converted to the bellowing equation.

$$\begin{aligned} &R_e[O_l(x)O_r*(x+\tau)] = \\ &R_e[O_l(x)]R_e[O_r(x+\tau)] - I_m[O_r(x)]I_m[O_r(x+\tau)] \end{aligned} \tag{8}$$

As a result, the computation of imaginary part is omitted and the Gaussian window is applied once to the result of the divider. <Table 2> lists the reduced hardware resources after optimization. The result shows that the reduced version saves around 55% of hardware resources in the numbers of LUTs and slices. In case of the number of flip flops, the reduction ratio reaches up to 75%. The hardware resources required in [11] are also expressed in terms of LUTs and flip flops. Here, 4-input LUTs and flip flops are required around 66,475 82,955, respectively. The proposed architecture can reduce flip-flops over 80% being compared with the real time stereo system in [11]. This shows the possibility that the proposed architecture can be integrated into one chip.

6. Conclusion

In this paper, high performance coprocessor architecture for dense disparity computation based on the local weighed phase-correlation algorithm is proposed. For parallel and efficient hardware implementation, some modifications on the original LWPC were necessary such as fixed-point data representation and decreasing the complexity of low-pass filters. For 2-dimensional convolution which is known as separable, two linear arrays working on pipelined mode, in which intermediate result from X array can be immediately fed into Y array without additional control. The important advantage is requiring much less hardware resources than non-separable filters of the same size. The simulation result shows the possibility that the proposed architecture can be integrated into one chip after reducing the computational complexity of G2-H2 filter block and the phase-correlation unit while maintaining the processing speed at video rate.

Acknowledgement

This work has been partly supported by the BK21 Research Center for Intelligent Mobile Software at Yonsei University in Korea. The author would like to thank Mr. Jin-Seok Heo for his help.

References

[1] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, Issue 8, pp.993-1008, Aug. 2003.

[2] H. Sunyoto, W. van der Mark, and D. M. Gavrila, "A comparative study of fast dense stereo vision algorithms," Proc. IEEE Intelligent Vehicles Symposium 2004, pp.319-324, 14-17 June 2004.

[3] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A stereo machine for video-rate dense depth mapping and its new applications" Proc. IEEE CVPR '96, pp.196-202, 1966.

[4] J. Y. Goulermas and P. Liatsis, "Feature-based stereo matching via coevolution of epipolar subproblems," Proc. Seventh International Conference on Image Processing And Its Applications, Vol.1, pp.23-27, 13-15 July 1999.

[5] D. Fleet, A. Jepson, and M. Jenkin, "Phase-based disparity measurement," CVGIP: Image Understanding, Vol.53, pp. 198-210, 1991.

[6] D. J. Fleet, "Disparity from local weighted phase-correlation," Proc. Int. Conf. on Systems, Man, and Cybernetics, Vol.1, pp.48-54, 1994.

[7] O. Faugeras, B. Hotz, H. Matthieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real time correlation-based stereo: Algorithm, implementations and applications," INRIA Technical Report 2013, 1993.

[8] L. Matthies, A. Kelly, T. Litwin and G. Tharp, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," Proc. Intelligent Vehicles '95 Symp., pp.66-71, 1995.

[9] S. Kimura, T. Kanade, and H. Kano, A. Yoshida, E. Kawamura, and K. Oda, "CMU Video-rate stereo machine," Proc. Mobile Mapping Symp., 1995.

[10] J. Woodfill and B. Von Herzen, "Real-time stereo vision on the PARTS recon-figurable computer," Proc. IEEE Workshop FPGAs for Custom Computing Machines, pp. 242-250, 1997.

[11] A. Darabiha, W. J. Mac Lean, and J. Rose, "Reconfigurable hardware implementation of a phase-correlation stereo algorithm," Machine Vision and Applications, Vol.17, No.2, pp.116-132, May 2006.

[12] C. Kuglin and D. Hines, "The phase correlation image alignment method," Proc. IEEE Int. Conf., Cybern. Soc., pp. 163-165, 1975.

[13] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," PAMI, Trans. on, Vol.13, Issue 9, pp. 891-906, Sept. 1991.

[14] C. Cuadrado, A. Zuloaga, J. L. Martin, J. Laizaro, and J. Jimenez, "Real-Time Stereo Vision Processing System in a FPGA IEEE Industrial Electronics," IECON 2006 - 32nd Annual Conference, pp.3455-3460, Nov. 2006.

[15] M. Gong and Y. H. Yang, "Near real-time reliable stereo matching using programmable graphics hardware," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. Vol.1, pp.924-931, 20-25 June 2005.

[16] IEEE, IEEE standard VHDL Language Reference Manual, std 1076-1993, New York, 1993.

[17] www.xilinx.com/ise/logic_design_prod/webpack.htm

김 정 길

e-mail : tetons@yonsei.ac.kr

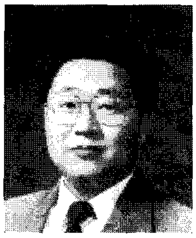
2003 M.S. Computer Science, Yonsei University.

2006 Ph.D. Computer Science, Yonsei University.

2006 Post-Doctor at the Dept. of Computer Science, Yonsei University.

2007~Current Research Prof. at the Dept. of Computer Science, Yonsei University.

Research areas : Computer Architecture, Multimedia Embedded Systems.



김 신 덕

e-mail : sdkim@yonsei.ac.kr

1981 B.S. Electronic Engineering, Yonsei University.

1987 M.S. Electrical Engineering, University of Oklahoma.

1991 Ph.D. Computer and Electrical Engineering, Purdue University.

1995~Current Prof. at the Dept. of Computer Science, Yonsei University.

Research areas : Advanced Computer Architectures, Parallel Processing Systems, Memory System Design, Agent-based Internet Computing.

Vason P. Srini

e-mail : srini@eecs.berkeley.edu

1969 B.E. Electrical Engineering, University of Madras.

1971 M.S. Electrical Engineering, Tennessee Technological University.

1980 Ph.D. Computer Science, University of Louisiana.

2004.~2006 Visiting Professor at the Yonsei University and ICU in Korea.

2007.~Current Research Prof. at UC Berkeley and Executive Director of Data Flux Systems Inc., Berkeley, CA. He has worked on parallel computer architectures, VLSI implementations, and software systems during the past three decades.

Research areas : Autonomous Ground Systems, Mobile Sensor Networks for Intelligent Transportation Systems, Parallel Embedded Systems.