2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Realtime and Robust Motion Tracking by Matched Filter on CMOS+FPGA Vision System

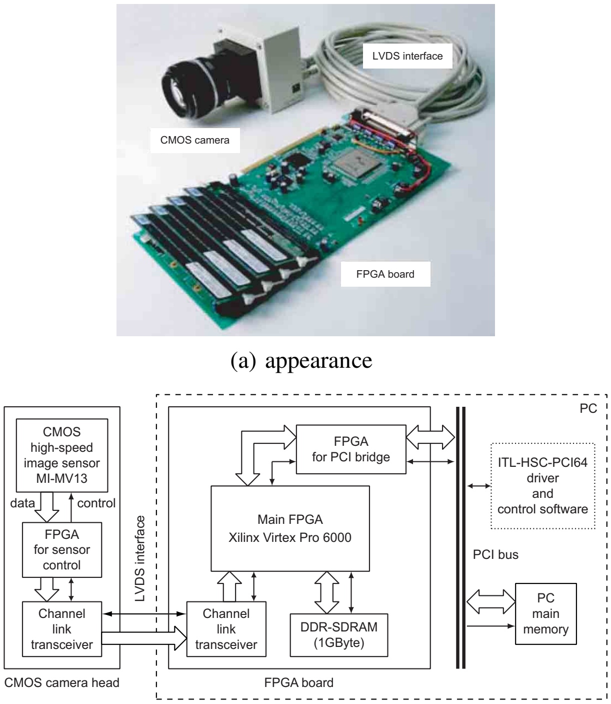
Kazuhiro Shimizu and Shinichi Hirai

Abstract— **This paper describes realtime and robust tracking of a planar motion target by matched filter implemented on the CMOS+FPGA vision system. It is required to obtain positional and angular signals around 1,000 Hz to control a mechanical system. A vision sensor must obtain visual features of a target object, synchronizing its sampling rate to the sampling rate of the control. The CMOS+FPGA vision system has been proposed to realize 1,000 Hz visual feedback. Matched filter can compute the position of a target robustly against occlusion, change of illumination, and background texture but requires much computation time since it includes 2D Fourier transform of images. Thus, matched filter algorithm is implemented on the system so that the matched filter can be performed in realtime. First we briefly introduce the CMOS+FPGA vision system. Second, we summarize the algorithm of matched filter to describe the design of the circuit performing matched filter on an FPGA. Finally, we show the experimental results of planar motion tracking by matched filter implemented on the system.**

I. INTRODUCTION

This paper describes realtime and robust tracking of a planar motion target by matched filter implemented on the CMOS+FPGA vision. It is required to obtain positional and angular signals around 1,000Hz to control a mechanical system. A vision sensor must obtain visual features of a target object, synchronizing its sampling rate to the sampling rate of the control. Thus, we need 1) image capturing over 1,000Hz with high resolution, 2) visual feature computation at the capturing rate, and 3) visual feature transmission to a control system with little delay. Visual feature extraction should be robust against disturbances in captured images.

In order to realize realtime computation of visual features, many vision algorithms have been implemented on LSIs. Logic circuits specialized for individual algorithms are designed and are implemented on LSIs. Such algorithms include Fourier transforms [1], Hough transforms [2], normalized correlations [3], [4], and stereo vision algorithms [5]. VLSIs composed of logic circuits and analog sensor circuits have been proposed [6], [7]. Fast computation is realized in such ASIC-based approach but requires tremendous time and staggering cost to design and implement logic circuits on ASICs. Ishikawa et al. have proposed a vision chip to realize 1,000Hz visual feedback [7]. This chip consists of image elements and each image element includes a photo detector and a computer with memory. Each image element has four data links connected to its neighboring elements. Namely, this chip is a full parallel computer distributed to an array of image detectors. Note that many computer scientists and engineers have studied full parallel computing over thirty years, finding that local computation can be realized easily in a full parallel computer but it is difficult to realize global computation. This suggests that this vision system can realize fast computation in local operations such as image moment calculation and image filters but can hardly implement global operations such as Fourier transform and Hough transform. Furthermore, this chip is a mixed analog/digital LSI: analog circuit for image capturing and digital circuit for computation. Any mixed analog/digital LSI has a drawback of interference between analog and digital circuits. Ishii et al. have proposed the Mm vision system to realize 1,000Hz visual feedback [8]. This system consists of a CMOS image sensor and an image region selector. A CMOS image sensor detects an image over 1M pixels over 1,000Hz. A selector specifies a rectangle region in a detected full image and transmits the specified region to a computer, where visual features are computed. This reduces the data transmission from an image detector to a computer and the time in image



(a) appearance

(b) architecture

Fig. 1. CMOS+FPGA vision

K. Shimizu is with the Graduate School of Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan rr006014@se.ritsumei.ac.jp

S. Hirai is with the Department of Robotics, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan hirai@se.ritsumei.ac.jp

1-4244-0602-1/07/$20.00 ©2007 IEEE.

feature computation. The computer specified the rectangle region in the next image capturing based on the obtained image features. This system works well unless obstacles occlude a target object, which often happen in real world.

A CMOS+FPGA vision system has been proposed to realize 1,000Hz visual feedback. The system consists of a CMOS image detector to capture images at 1,000Hz and an FPGA to compute image features at this sampling rate. Note that analog circuits in the image detector are separated from digital circuits in the FPGA. Many vision algorithms have been implemented on FPGAs: not only local operations but global operations such as 2D discrete cosine transform [9], image restortion based on convolution [10], and Hough transform using CORDIC [11]. This suggests that the system can realize local and global operations by implementing them on an FPGA.

In this paper, matched filter algorithm is implemented on the CMOS+FPGA vision system to realize realtime and robust tracking of a planar motion target. Matched filter applies 2D Fourier transform of images to compute the position of a target robustly against occlusion, change of illumination, and background texture. Two-dimensional Fourier transform requires much computation time in CPUs, resulting that PCs cannot perform matched filter in realtime. Fortunately, 2D Fourier transform has high parallelism, suggesting that implementing 2D Fourier transform on FPGAs reduced the computation time and making matched filter to be performed in realtime. First we briefly introduce the CMOS+FPGA vision system. Second, we summarize the algorithm of matched filter to describe the design of the circuit performing matched filter on an FPGA. Then, we show the experimental results of planar motion tracking by matched filter implemented on the CMOS+FPGA vision system.

II. CMOS+FPGA VISION SYSTEM

A. Concept

The CMOS+FPGA vision system consists of a CMOS image detector to capture successive images at 1,000Hz and an FPGA to compute image features at this sampling rate. Successive images captured by the CMOS imager are sent to the FPGA, where a vision algorithm circuit is implemented, via high-speed data connection, LVDS. Logic circuits in the FPGA compute image features of each captured image. Image features are sent to a computer using PCI bus. Note that data size of image features is quite less than the size of each captured image, enabling the vision system to send data to a computer via PCI bus.

Analog circuits in the CMOS image detector and digital circuits in the FPGA are separated in this CMOS+FPGA vision system, avoiding interference between analog and digital circuits. Since analog circuit requires more current than digital circuit, the current in analog circuits often disturbs digital circuits, if these circuits are close to one another. The CMOS+FPGA vision system excludes the mixture of analog and digital circuits, yielding reliable performance.

B. Architecture

We have developed a CMOS+FPGA vision system, ITL-HSC-AD, fabricated by Image Technology Laboratory Corporation. Figure 1 describes the vision system. Figure 1-(a) shows the appearance of ITL-HSC-AD. As shown in the figure, this system consists of a CMOS camera head, an LVDS interface, and an FPGA board. The camera head has a CMOS image detector, Micron Imaging MI-MV13, which can capture 1280×504 pixels at 1,000 frame per second. The FPGA board includes an FPGA, Xilinx Vertex Pro 6000, where we can implement 6 million system gates. Figure 1-(b) describes the architecture of the CMOS+FPGA vision system. Images captured by the CMOS camera are sent to the FPGA via LVDS interface. Logic circuits for vision algorithms are implemented on the FPGA. Each logic circuit processes the sent images in realtime, using DDR-SDRAMs for memory if necessary. A PCI bridge provides the data communication between the FPGA and PCI bus.

We use Xilinx ISE Foundation Verilog-HDL for the design of logic circuits for vision algorithms and Mentor Graphics ModelSim SE for the simulation of designed circuits.

III. MATCHED FILTER

A. Algorithm

This section describes the implementation of matched filter on the CMOS+FPGA vision. The matched filter [12] can detect the translation between two images robustly against the background and illumination changes. This algorithm includes two global operations: 2D-FFT and 2D-IFFT. These operations require much computation time but have high parallelism, suggesting that implementing the matched filter algorithm on an FPGA reduces the computation time.

Let $g_{\rm ref}(x,y)$ and $g_{\rm inp}(x,y)$ be reference and input images. Let $G_{\rm ref}(\xi,\eta)$ and $G_{\rm inp}(\xi,\eta)$ be their 2D Fourier transforms. Correlation function $c(x,y)$ can be obtained by applying 2D inverse FFT to the complex quotient given by

$$C(\xi,\eta) = \frac{G_{\rm inp}(\xi,\eta)}{G_{\rm ref}(\xi,\eta)}.$$

It has been proved that when the reference and input images are identical with translation given by (x_o,y_o), say, $g_{\rm inp}(x,y) = g_{\rm ref}(x-x_o, y-y_o)$ is satisfied, correlation function coincides to a delta function $\delta(x-x_o, y-y_o)$. Thus, searching the maximum value of the correlation function, we can detect the translation between the reference and input images. When an input image is blurred by change of illumination, occlusion, or background texture, the correlation function no longer coincides to a delta function. But, as long as the correlation function takes its maximum at (x_o,y_o), we can detect the translation robustly.

B. Circuit Design

Figure 2 shows the overview of a matched filter circuit, consisting of 1) two-dimensional FFT module, 2) complex quotient module, 3) two-dimensional IFFT module, 4) peak detection module, and 5) reference FFT buffer module. The reference FFT buffer module stores the Fourier transform

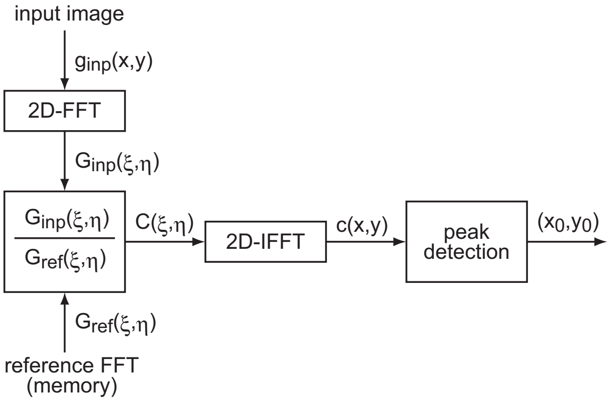
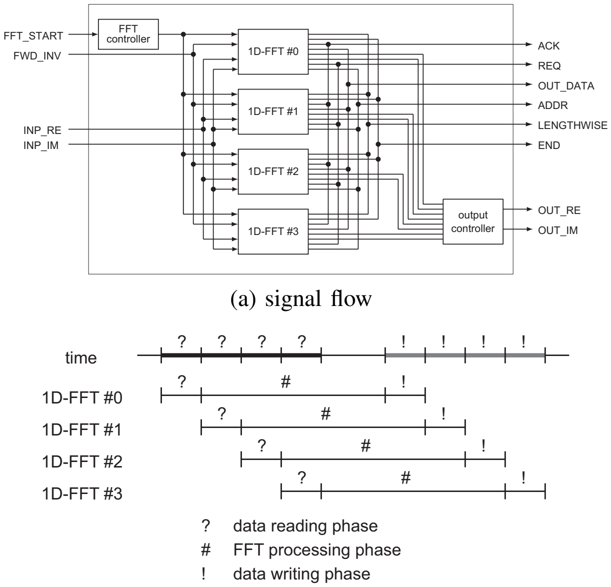


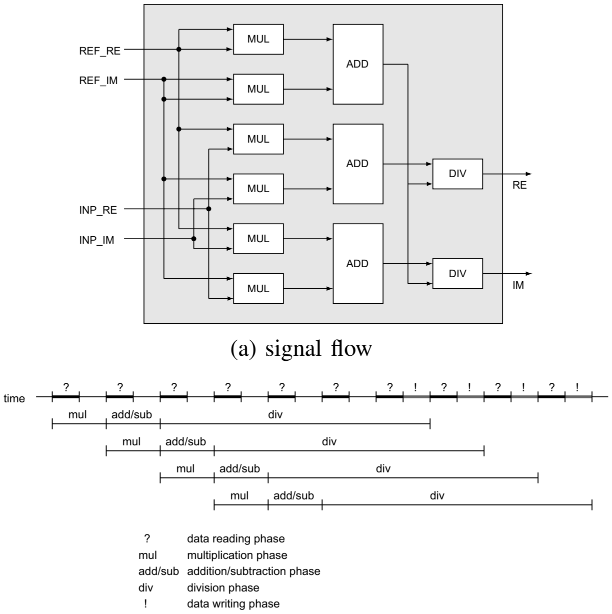
Fig. 2. Overview of matched filter circuit



(a) signal flow

(b) time chart

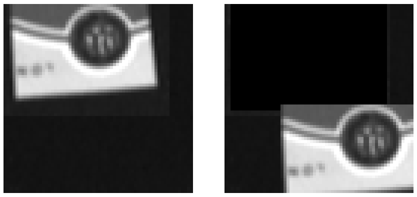
Fig. 3. Pipelined 2D FFT circuit



(a) signal flow

(b) time chart

Fig. 4. Circuit to compute complex quotient



(a) reference (b) input

Fig. 5. Reference and input images used in simulation

of a reference image, $G_{\text{ref}}(\xi,\eta)$, in the DDR-SDRAM beforehand. Input image $g_{\text{inp}}(x,y)$ is sent to the 2D-FFT module to compute its Fourier transform, $G_{\text{inp}}(\xi,\eta)$. Concurrently, the complex quotient module calculates $C(\xi,\eta) = G_{\text{inp}}(\xi,\eta)/G_{\text{ref}}(\xi,\eta)$. Two-dimensional IFFT module computes the 2D-IFFT of the quotient to obtain correlation function $c(x,y)$. Let $c_{\text{peak}} = c(x_o,y_o)$ be the maximum value of function $c(x,y)$. Peak detection module searches the maximum value c_{peak} and coordinates (x_o,y_o).

Figure 3 shows a pipelined 2D FFT module. Recall that 2D FFT can be realized by 1D FFT along rows and 1D FFT along columns. We designed a 1D FFT circuit based on the Xilinx 64-point FFT IP core and dual port RAMs, making the image size be 64×64 pixels. As shown in the figure, we applied four 1D-FFT circuits in the 2D FFT module to speed up the computation. Figure 3-(a) details the flow of signals in the 2D FFT module. Signal FWD_INV specifies if this module computes FFT or IFFT. Signals INP_RE and INP_IM denote the real and imaginary part of an input signal. Signals OUT_RE and OUT_IM denote the real and imaginary part of an output signal. Figure 3-(b) shows the time chart during the computation in the module. Four 64-point data are sent to four 1D-FFT circuits one by one to perform pipelined computation of the four 1D-FFTs.

Figure 4 shows a circuit design for the complex quotient module. Figure 4-(a) details the flow of signals in the complex quotient module. Signals REF_RE and REF_IM denote the real and imaginary part of Fourier transform $G_{\text{ref}}(\xi,\eta)$. Signals INP_RE and INP_IM describe the real and imaginary part of Fourier transform $G_{\text{inp}}(\xi,\eta)$. Signals RE and IM denote the real and imaginary part of the complex quotient. This module consists of adder, subtracter, multiplier, and divider. Figure 4-(b) shows the time chart during the computation in the module. Since a divider yields 54-clock delay, we apply pipeline processing to reduce the total computation time. Namely, the module controls the signal flow so that the data input and the result output alternate. We have used hardware multipliers provided in Xilinx Vertex Pro 6000.

We have simulated the behavior of the designed circuit using the ModelSIM SE. We have applied the post-translate model simulation to take the delay of IP cores into consideration. Figure 5 shows 64× 64 reference and input images

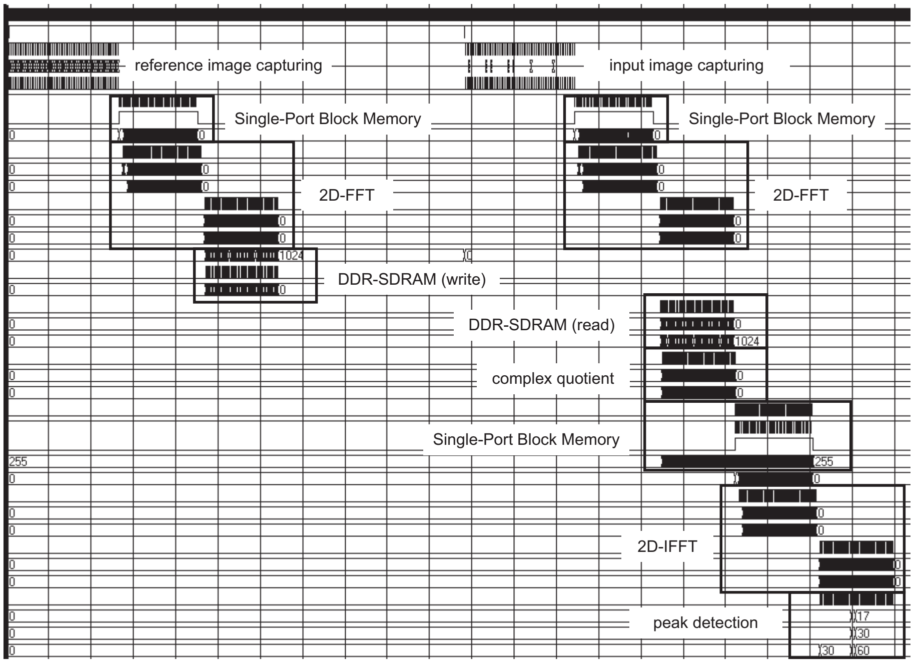


Fig. 6. Post-translate Verilog model simulation of matched filter circuit

TABLE I

CONSUMPTION FPGA RESOURCES IN MATCHED FILTER CIRCUIT

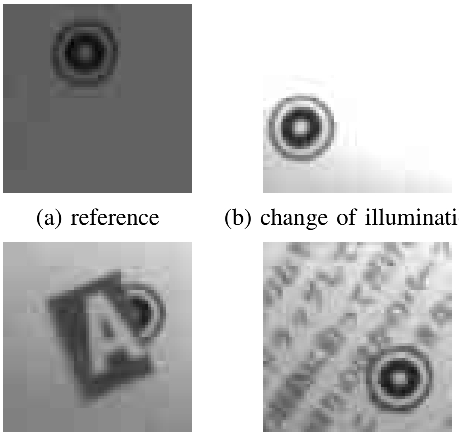
2D-FFT		
Time	0.181 ms	
Number of Slices	5149 out of 33792	15 %
Number of Block RAMs	8 out of 144	5 %
Complex quotient		
Time	0.086 ms	
Number of Slices	4614 out of 33792	13 %
Number of Multipliers	6 out of 144	4 %
2D-IFFT		
Time	0.181 ms	
Number of Slices	5149 out of 33792	15 %
Number of Block RAMs	8 out of 144	5 %
Peak detection		
Time	0.086 ms	
Number of Slices	111 out of 33792	1 %
Total		
Time	0.647 ms	
Number of Slices	15070 out of 33792	44 %
Number of Block RAMs	124 out of 144	86 %
Number of Multipliers	7 out of 144	4 %

used in the simulation. The two images are identical with translation given by $(x_o, y_o) = (17, 30)$. Figure 6 shows a simulation result. First, the CMOS camera captures a reference image, then the 2D FFT module computes the 2D FFT of the image the FFT result is stored in DDR-SDRAM. Single-port block memory in the FPGA stores the image temporary to ensure the timing between image capturing and FFT computation. Second, the CMOS camera captures an input image before its 2D FFT is computed. During the computation of input image FFT, reference image FFT is read from the DDR-SDRAM. The complex quotient between the two FFTs is computed in parallel to store the result in single-port block memory. Then, the IFFT of the quotient is computed before detecting the peak of the IFFT. We find that output signals of peak detection module are 17 and 30, suggesting that translation given by $(x_o, y_o) = (17, 30)$ is successfully obtained in this simulation.

Table I summarizes the simulation result. It takes 0.181 ms to compute the 2D Fourier transform of a 64×64 pixel image. It takes 0.086 ms to compute the complex quotient of two 64×64 Fourier transforms. It takes 0.086 ms to detect the peak of a 64×64 correlation function. It totally takes 0.647 ms to perform matched filter between two 64×64 images. Since we need a time to synchronize the image capturing and the designed circuit, the total time is more than a simple sum of the times in individual modules. The table shows FPGA resources (slices, block RAMs, and multipliers) consumed in the designed circuit. A slice implies a logic element in the FPGA. Two-dimensional FFT, complex quotient, two-dimensional IFFT, and peak detection modules consume 15 %, 13 %, 15 %, and 1 % of the whole slices. The designed circuit consumes 44 % slices totally, which exceeds a simple sum of the slices in individual modules since connection among modules consumes slices. The designed circuit totally consumes 86 % of block RAMs and 4 % of hardware multiplies. This suggests that consumption of block RAMs may be a bottleneck against the matched filter between two 256×256 images. We should redesign a matched filter circuit to reduce the consumption of block RAMs for matched filter between 256×256 images.

IV. TRACKING EXPERIMENTS

This section shows experimental results of the tracking performed by matched filter implemented on the CMOS+FPGA vision system. A point-symmetric marker is



(a) reference (b) change of illumination

(c) occlusion (d) background texture

Fig. 7. Images for matched filter on FPGA

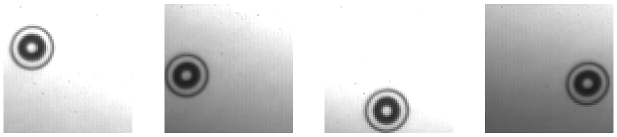


Fig. 8. Images with change of illumination

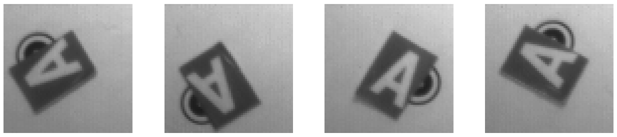


Fig. 9. Images with occlusion

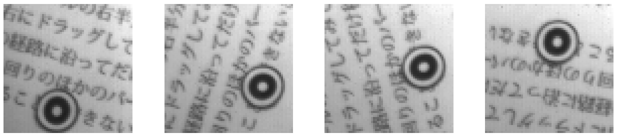
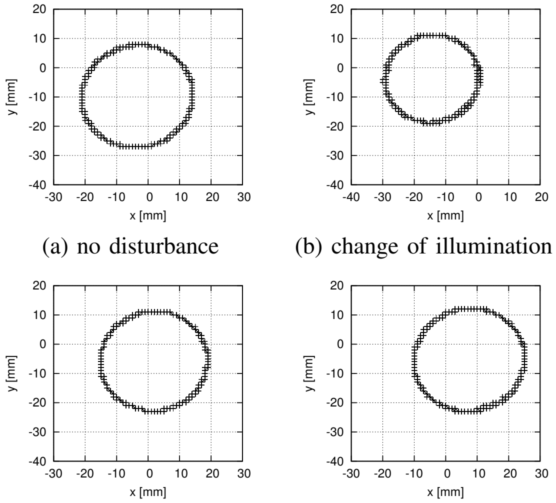


Fig. 10. Images with background texture



(a) no disturbance (b) change of illumination

(c) occlusion (d) background texture

Fig. 11. Tracking results by matched filter on FPGA

attached to the top of a rigid link rotating around a joint fixed on space. We have tested if the matched filter circuit can detect a reference image given in Figure 7-(a) in a sequence of input images with disturbances. Figure 7-(b) shows an input image with the change of illumination. As shown in Figure 8, the illumination varies during the rotation of a point-symmetric marker. Figure 7-(c) shows an input image with occlusion. As shown in Figure 9, the marker is occluded by an obstacle during the rotation of the marker. Figure 7-(d) shows an input image with background texture. As shown in Figure 10, the background of the marker is blurred by texture. Figure 11-(a) plots the result of the tracking in a sequence of images without any disturbance. Figure 11-(b) plots the result of the tracking in a sequence of images with the change of illumination. Figure 11-(c) plots the result of the tracking in a sequence of images with occlusion. Figure 11-(d) plots the result of the tracking in a sequence of images with background. The above figures show that the matched filter implemented on the CMOS+FPGA vision can detect the position of a reference image robustly against the change of illumination, occlusion, and background.

Figure 12 shows a sequence of successive input images disturbed by background texture and the change of illumination. Figure 13 shows a sequence of successive input images disturbed by background texture and occlusion. Figure 14-(a) plots the result of the tracking in a sequence of images disturbed by background texture and the change of illumination. The matched filter implemented on the CMOS+FPGA vision succeeded to detect the position robustly despite of such combined disturbances. Computed x- and y-coordinates during tracking plotted in Figure 15-(a) also prove the successful detection. Figure 14-(b) plots the result of the tracking in a sequence of images disturbed by background texture and occlusion. The matched filter often failed to detect the position against such combined disturbances. But, the detection can be successful even after the failure since the position is computed at each sampling time. Actually, the failure happens intermittently as shown in Figure 15-(b), where computed x- and y-coordinates during tracking are plotted. This suggests that the failure can be detected by checking the rate of change or provides little harmful influence to control systems.

V. CONCLUDING REMARKS

This paper has shown realtime and robust tracking of a planar motion target by matched filter implemented on the CMOS+FPGA vision. We showed the implementation of matched filter on the system and found that it takes about 0.647 ms for the matched filter between two 64×64 pixel images. Then, we tested the planar motion tracking performed by the implemented matched filter. We find that the matched filter on the CMOS+FPGA vision system can detect the position of a reference image robustly against the change of illumination, occlusion, and background. On the other side, the matched filter failed to detect the position along a sequence of images disturbed by both background texture and occlusion.

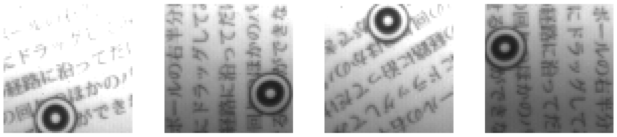


Fig. 12. Images with background texture and change of illumination

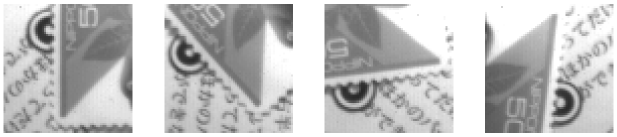
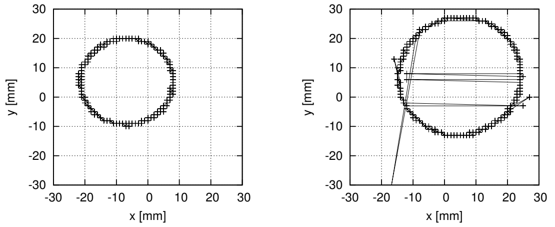
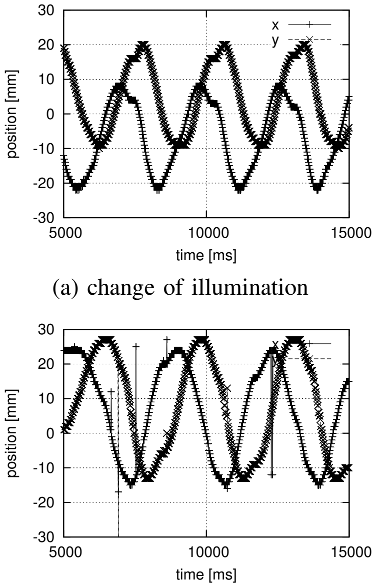


Fig. 13. Images with background texture and occlusion



(a) change of illumination (b) occlusion

Fig. 14. Tracking results along images under background texture



(a) change of illumination

(b) occlusion

Fig. 15. Computed coordinates during tracking along images under background texture

Ongoing issues include 1) implementation of matched filter between 256×256 images and 2) visual feedback control using the matched filter circuit. Matched filter between 256×256 images requires to use external DDR-SDRAMs. We are redesigning the matched filter circuit. Visual feedback using the implemented matched filter can increase the robustness in control. We are going to test the robustness in the control of flexible links. Future issues include 1) implementation of rotation-invariant matched filter algorithm on the CMOS+FPGA vision systems to detect both translation and rotation of a planar motion object and 2) implementation of SNAKE to detect a deformable contour in realtime.

ACKNOWLEDGMENTS

This research was supported in part by the Ritsumeikan University 21st Century COE program "Micro Nanoscience Integrated Systems". We deeply appreciate helpful advices on ITL-HSC-AD by the members of Image Technology Laboratory Corporation.

REFERENCES

[1] C. D. Thompson, *Fourier Transforms in VLSI*, IEEE Trans. on Computers, Vol.C-32, No.11, pp. 1047–1057, 1983.

[2] M. Maresca, M. Lavin, and H. Li, *Parallel Hough Transform Algorithms on Polymorphic Torus Architecture*, S. Levialdi eds., *Multicomputer Vision*, Academic Press, pp. 9–21, 1988.

[3] H. Inoue, T. Tachikawa, and M. Inaba, *Robot Vision System with a Correlation Chip for Real-time Tracking, Optical Flow and Depth Map Generation*, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1621–1626, Nice, May, 1992.

[4] A. Bugeja, and W. Yang, *A Reconfigurable VLSI Coprocessing System for the Block Matching Algorithm*, IEEE Trans on VLSI Systems, Vol.5, No.3, pp. 329–337, 1995.

[5] M. Hariyama, T. Takeuchi, and M. Kameyama, *VLSI Processor for Reliable Stereo Matching Based on Adaptive Window-Size Selection*, Proc. 2001 IEEE Int. Conf. on Robotics and Automation, pp. 1168–1173, Seoul, May, 2001.

[6] J.-E. Eklund, C. Svensson, and A. Aström, *VSLI Implementation of a Focal Plane Image Processor – A Realization of the Near-Sensor Image Processing Concept*, IEEE Trans. on VLSI Systems, No.4, Vol.3, pp. 322–335, 1996.

[7] I. Ishii, Y. Nakabo, and M. Ishikawa, *Target Tracking Algorithm for 1ms Visual Feedback System using Massively Parallel Processing Vision*, Proc. 1996 IEEE Int. Conf. on Robotics and Automation, pp. 2309–2314, Minneapolis, May, 1996.

[8] T. Komuro, I. Ishii, M. Ishikawa, and A. Yoshida, *A Digital Vision Chip Specialized for High-speed Target Tracking*, IEEE transaction on Electron Devices, Vol.50, No.1, pp. 191–199, 2003.

[9] R. Woods, D. Trainor, and J.-P. Heron, *Applying an XC6200 to Real-Time Image Processing*, IEEE Design & Test of Computers, Vol.15, No.1, pp. 30–38, 1998.

[10] S. O. Memik, A. K. Katsaggelos, and M. Sarrafzadeh, *Analysis and FPGA Implementation of Image Restortion under Resource Constraints*, IEEE Trans. on Computers, Vol.52, No.3, pp. 390–399, 2003.

[11] D. D. S. Deng, and H. ElGindy, *High-speed Parameterisable Hough Transform Using Reconfigurable Hardware*, Proc. Pan-Sydney area Workshop on Visual Information Processing, pp. 51–57, 2001.

[12] G. L. Turin, *An Introduction to Digital Matched Filters*, Proceedings of the IEEE, Vol.64, No.7, pp. 1093–1112, 1977.

[13] S. S. Ge, T. H. Lee,, and G. Zhu, *Improving Joint PD Control of Single-link Flexible Robots by Strain/Tip Feedback*, Proc. of IEEE International Conference on Control Applications, Dearborn, pp. 965–969, 1996.

[14] F. Matsuno, T. Ohno, and Y. V. Orlov, *Proportional Derivative and Strain (PDS) Boundary Feedback Control of a Flexible Space Structure with a Closed-Loop Chain Mechanism*, Automatica, 38(7), pp. 1201–1211, 2002.