

CMOS+FPGA Vision System for Visual Feedback of Mechanical Systems

Kazuhiro Shimizu and Shinichi Hirai
*Dept. Robotics, Ritsumeikan Univ.
Kusatsu, Shiga 525-8577, Japan*

Abstract—This paper describes a 1,000Hz visual feedback using 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.

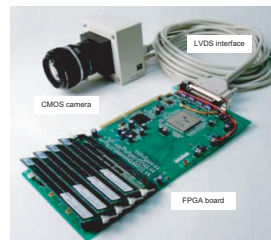
We propose the CMOS+FPGA vision system to realize 1,000Hz visual feedback. This system consists of a CMOS image detector to capture images at 1,000Hz and an FPGA to compute image features at this sampling rate. First we introduce the CMOS+FPGA vision system. Second, we show the implementation of the computation of the image gravity center. Then, we show the implementation of matched filter. Finally, we apply the image gravity center circuit to the visual control of a flexible link.

Index Terms—vision, realtime, visual feedback, CMOS image sensor, FPGA.

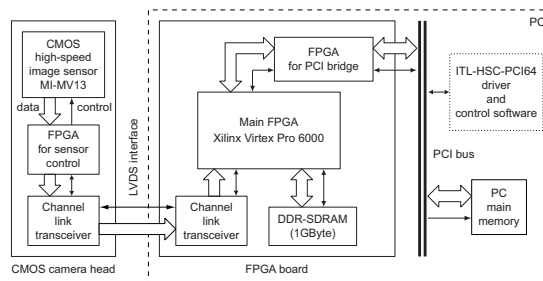
I. INTRODUCTION

This paper describes a 1,000Hz visual feedback using 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.

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



(a) appearance



(b) architecture

Fig. 1. CMOS+FPGA vision

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 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.

We propose a CMOS+FPGA vision system to realize 1,000Hz visual feedback. This 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 restoration based on convolution [10], and Hough transform using CORDIC [11]. This suggests that our system can realize local and global operations by implementing them on an FPGA. Detected image features can be sent to a computer via PCI bus at this sampling rate. In this paper, first we introduce the CMOS+FPGA vision system. Second, we show the implementation of the computation of the image gravity center. Then, we show the implementation of matched filter. Finally, we apply the image gravity center circuit to the visual control of a flexible link.

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

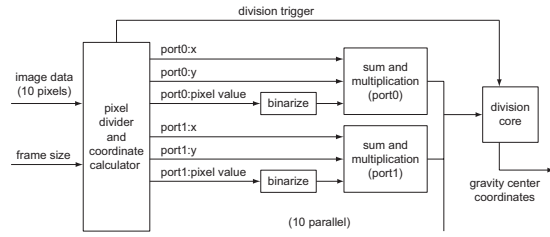


Fig. 2. Circuit to compute image gravity center



Fig. 3. Input image

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. COMPUTATION OF IMAGE GRAVITY CENTER

We have implemented a logic circuit to compute the image gravity center on the FPGA. Figure 2 shows the design of logic circuit. The CMOS camera captures 1280×504 pixel images at 1,000 fps. Each pixel has 10 bit grayscale level. The CMOS camera sends 10 pixel data to the FPGA at each clock. Thus, we have designed 10 parallel logic circuit to compute the image gravity center at the minimum latency. In this design, computation finishes in the time needed for one division. The circuit consumes 8% of total gates on the FPGA. Simulation shows that the circuit is driven at 73 MHz at a maximum. In the following experiments, we drive the circuit at 66 MHz. Then, the latency is 0.69 μ s.

We have applied the CMOS+FPGA vision system to the detection of an LED rotating at a constant angular velocity. A rigid link is fixed to the shaft of a DC motor, which rotates at a constant angular velocity according to the voltage applied to it. An LED is attached to the top of the link. We had an experiment to detect an LED rotating at 600 rpm. The exposure time of the CMOS camera is 0.1 ms.

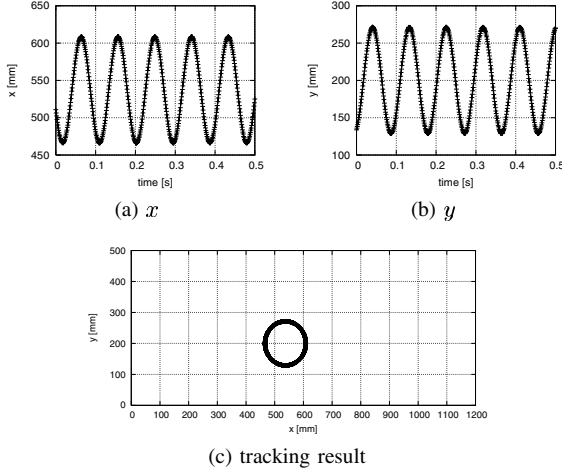


Fig. 4. Experimental result : detection of LED rotating at 600 rpm

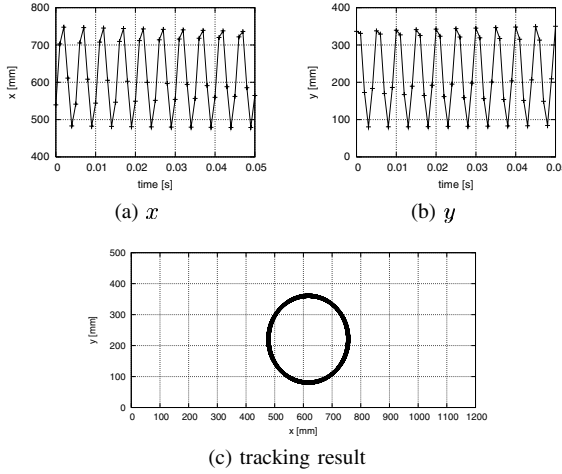


Fig. 5. Experimental result : detection of LED rotating at 12,000 rpm

Figure 3 shows an input image of the CMOS camera. Figure 4 shows an experimental result of the detection. Figures 4-(a) and (b) show the x- and y- coordinates of the image gravity center. The two coordinates are sinusoidal, implying that the image gravity center follows a circle, as shown in Figure 4-(c). In addition, it takes 0.1 s per one rotation, which agrees with the angular velocity of the rotating LED. Thus, we find that the CMOS+FPGA vision system detects the LED position successfully.

Figure 5 shows an experimental result of the detection of an LED rotating at 12,000 rpm. The exposure time of the CMOS camera is 0.03 ms. Figure 5-(c) shows a path of the image gravity center, implying that the image gravity center follows a circle. Figures 5-(a) and (b) show the x- and y- coordinates of the image gravity center. As shown in the figures, it takes 0.005 s per one rotation, which agrees with the angular velocity of the rotating LED. Moreover,

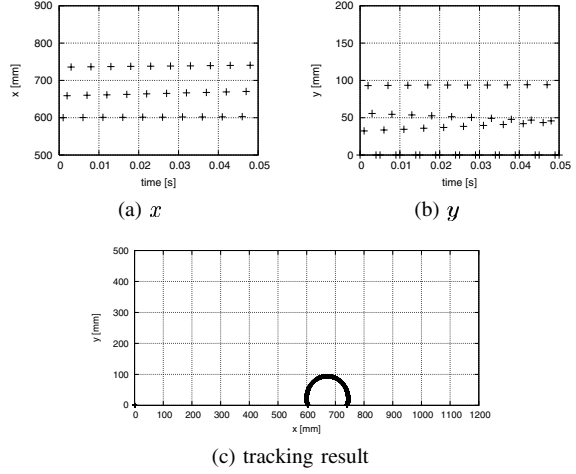


Fig. 6. Experimental result : detection of LED rotating at 12,000 rpm occluded by obstacle

we find 5 data points during one rotation, suggesting that the vision system detects the image gravity center at every 1 ms. Thus, we find that the CMOS+FPGA vision system detects the LED position successfully.

Figure 6 shows an experimental result of the detection of an LED occluded by an obstacle. The LED is rotating at 12,000 rpm but a horizontal obstacle hides the LED periodically. The exposure time of the CMOS camera is 0.03 ms. Figure 6-(c) shows a path of the image gravity center, suggesting that the image gravity center follows a circle within the camera view. Figures 6-(a) and (b) show the x- and y- coordinates of the image gravity center. As shown in the figures, it takes 0.005 s per one rotation, which agrees with the angular velocity of the rotating LED. Moreover, we find 5 data points during one rotation, suggesting that the vision system detects the image gravity center at every 1 ms. Thus, we find that the CMOS+FPGA vision system detects the LED position successfully despite of the occlusion.

IV. MATCHED FILTER

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

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

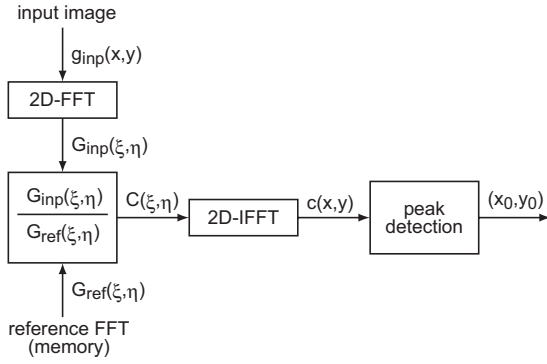
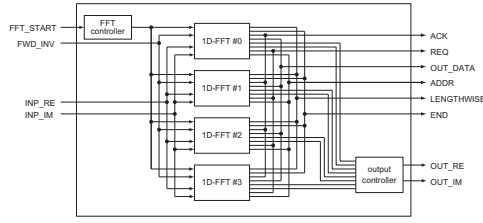
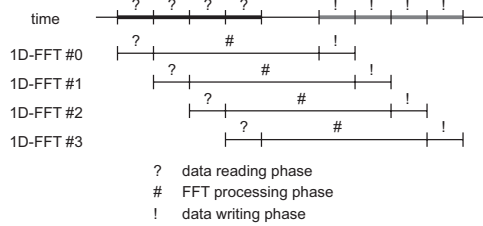


Fig. 7. Overview of matched filter circuit



(a) signal flow

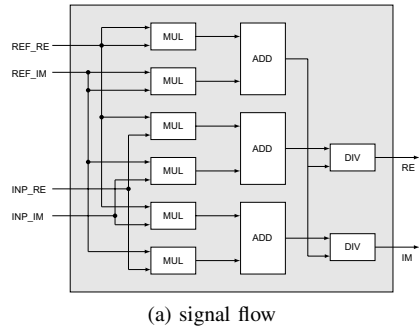


(b) time chart

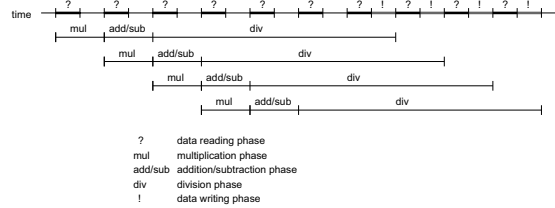
Fig. 8. Pipelined 2D FFT circuit

It has been proved that when the reference and input images are identical with translation given by (x_o, y_o) , say, $g_{inp}(x, y) = g_{ref}(x - x_o, y - y_o)$ 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.

Figure 7 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 of a reference image, $G_{ref}(\xi, \eta)$, in the DDR-SDRAM beforehand. Input image $g_{inp}(x, y)$ is sent to the 2D-FFT module to compute its Fourier transform, $G_{inp}(\xi, \eta)$. Concurrently, the complex quotient module calculates $C(\xi, \eta) = G_{inp}(\xi, \eta)/G_{ref}(\xi, \eta)$. Two-dimensional IFFT module computes the 2D-IFFT of the quotient to obtain correlation function $c(x, y)$. Let $c_{peak} = c(x_o, y_o)$



(a) signal flow



(b) time chart

Fig. 9. Circuit to compute complex quotient

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 8 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 256-point FFT IP core and dual port RAMs, making the image size be 256×256 pixels. As shown in the figure, we applied four 1D-FFT circuits in the 2D FFT module to speed up the computation. Figure 8-(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 8-(b) shows the time chart during the computation in the module. It takes 2.01 ms to compute the Fourier transform of a 256×256 pixel image.

Figure 9 shows a circuit design for the complex quotient module. Figure 9-(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 $G_{ref}(\xi, \eta)$. Signals INP_RE and INP_IM describe the real and imaginary part of Fourier transform $G_{inp}(\xi, \eta)$. Signals RE and IM denote the real and imaginary part of the complex quotient. This module consists of adder, subtractor, multiplier, and divider. Figure 9-(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. It takes 0.49 ms to compute the complex

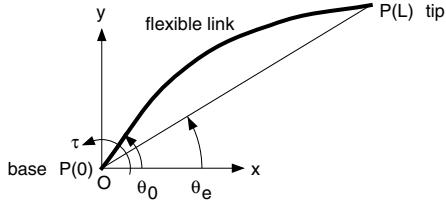


Fig. 10. Control of flexible link

quotient of two 256×256 Fourier transforms.

The matched filter mainly includes one 2D-FFT, one complex quotient calculation, and one 2D-IFFT in serial. This implies that it takes about 4.51 ms for the matched filter between two 256×256 pixel images.

V. CONTROL OF FLEXIBLE LINK

This section describes the control of a flexible link. Control of flexible manipulators has been extensively studied in the past decades. Many of the proposed control laws use strain signals to detect the deformation of a flexible manipulator [13], [14]. Strain signals are often noisy during the dynamic motion of a manipulator and tend to be blurred by an electromagnetic field due to actuators and their driving circuits. In this section, we employ visual signals instead of strain signals to detect the deformation of a flexible link.

Let θ_0 be the angle from the horizon at base point $P(0)$, where a torque is applied to a flexible link, as illustrated in Figure 10. Let θ_e denotes the angle from the horizon of vector connecting base point $P(0)$ and tip point $P(L)$. This angle is referred to as the *tip angle*. Note that the tip angle can be computed from the tip position observed by a vision system. Goal of a control law is to guide the tip angle θ_e to its desired angle θ_d and to suppress the vibration of the link. Let us first introduce the following PID term with respect to the tip angle so that the tip angle can be guided to its desired angle:

$$\tau_{\text{tip}} = K_p^{\text{tip}}(\theta_d - \theta_e) + K_i^{\text{tip}} \int_0^t (\theta_d - \theta_e) dt - K_d^{\text{tip}} \dot{\theta}_e,$$

where K_p^{tip} , K_i^{tip} , and K_d^{tip} denote proportional, integral, and derivative gains of tip angle feedback. Due to the delay between tip angle phase and base angle phase originated from the flexibility of the link, we should select a small value for derivative gain K_d^{tip} . Thus, we introduce an additional damping term given by

$$\tau_{\text{base}} = -K_d^{\text{base}} \dot{\theta}_0,$$

where K_d^{base} stands for a derivative gain of base angle feedback. Let us introduce the following term to suppress the vibration of the link:

$$\tau_{\text{sup}} = K_p^{\text{sup}}(\theta_e - \theta_0),$$

where K_p^{sup} represents a proportional gain. As a result, a control law for a flexible link is described as

$$\tau = \tau_{\text{tip}} + \tau_{\text{base}} + \tau_{\text{sup}}.$$

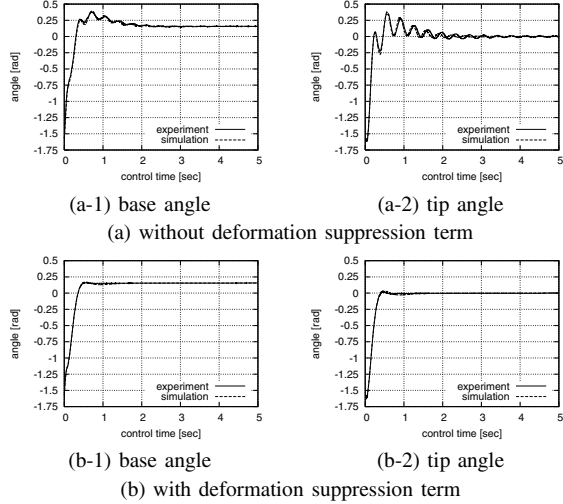


Fig. 11. Experimental result ($K_p^{\text{tip}} = 0.35$, $K_i^{\text{tip}} = 0.58$, $K_d^{\text{tip}} = 0.02$, $K_d^{\text{base}} = 0.04$, $K_p^{\text{sup}} = 0.8$)

Note that this control law includes no geometrical or physical parameters on a flexible link.

Let us apply the CMOS+FPGA vision system to experimentally verify the proposed control law of a flexible link. A flexible link composed of a spring steel of 0.5mm thickness is driven by an AC servo motor at the base point. An LED is attached to the tip point of the link. The CMOS+FPGA vision system detects the image gravity center of the LED using the logic circuit presented in the previous section. The detected image gravity center is feedback to a PC to control the AC servo motor according to the proposed control law. Experimental results are plotted in Figure 11. The desired angle θ_d is constantly equal to 0. Figure 11-(a) shows a result without the vibration suppress term. As plotted in Figures 11-(a-1) and (a-2), tip angle θ_e vibrates even after base angle θ_0 converges to a certain angle. Figure 11-(b) shows a result with the vibration suppress term. As plotted in Figures 11-(b-1) and (b-2), tip angle θ_e converges to its desired value as base angle θ_0 converges to a certain angle.

Figure 12 shows experimental results when an additional load is attached at the tip. Note that we have used the same gains as the previous experiment. As shown in the figures, the tip angle converges to its desired value against the load at the tip. This robustness comes from the proposed control law without physical parameters of the flexible link. Figures 13-(a) through (j) demonstrate the motion of the flexible link. From the above experiment, we find that the proposed vibration suppress term is effective to the control of a flexible link.

VI. CONCLUDING REMARKS

This paper has shown a 1,000Hz visual feedback using the CMOS+FPGA vision. First we introduced the

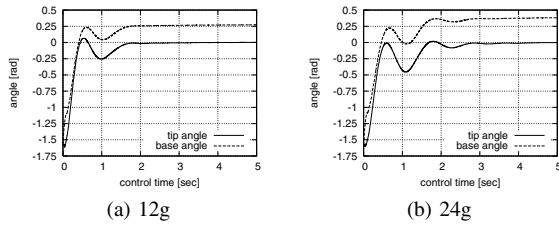


Fig. 12. Experimental result with additional load at tip

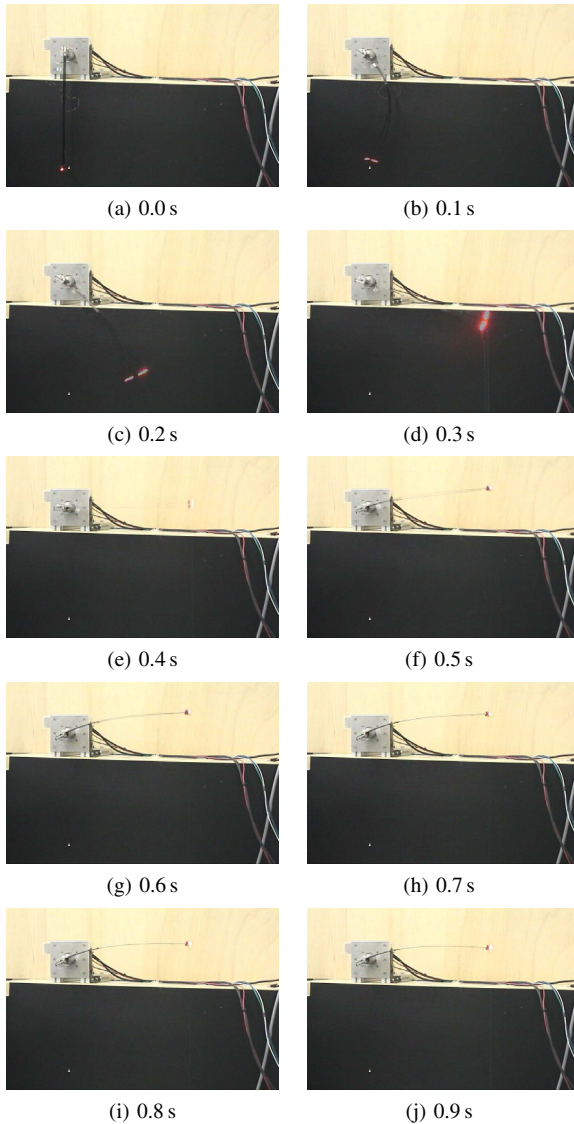


Fig. 13. Motion of flexible link with 24g load at tip

CMOS+FPGA vision system, consisting of a CMOS image sensor and an FPGA. Second, we described the implementation of the computation of the image gravity center. Our design computes the image gravity center $0.69 \mu\text{s}$ after the image capturing. Then, we show the implementation of the matched filter. We find that it takes about 4.51 msec for the matched filter between two 256×256 pixel images. Finally, we applied the image gravity center circuit to the control of a flexible link. We find that a simple control law without physical parameters of the link works well against a load at the tip.

ACKNOWLEDGEMENT

This research was supported in part by the Ritsumeikan University 21st Century COE program “Micro Nanoscience Integrated Systems”.

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. Levidali 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, *VLSI 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 Restoration 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.