Paper
Int'l J. of Aeronautical & Space Sci. 17(2), 232–239 (2016)
DOI: http://dx.doi.org/10.5139/IJASS.2016.17.2.232

IJASS
International Journal of Aeronautical and Space Sciences

FPGA based HW/SW co-design for vision based real-time position measurement of an UAV

Young Sik Kim*, Jeong Ho Kim, Dong In Han**, Mi Hyun Lee**, Ji Hoon Park****

and Dae Woo Lee***

Department of Aerospace Engineering Pusan National University, Jangjeon 2-dong, Geumjeong-gu, Busan 46241, Republic of Korea

Abstract

Recently, in order to increase the efficiency and mission success rate of UAVs (Unmanned Aerial Vehicles), the necessity for formation flights is increased. In general, GPS (Global Positioning System) is used to obtain the relative position of leader with respect to follower in formation flight. However, it can't be utilized in environment where GPS jamming may occur or communication is impossible. Therefore, in this study, monocular vision is used for measuring relative position. General PC-based vision processing systems has larger size than embedded systems and is hard to install on small vehicles. Thus FPGA-based processing board is used to make our system small and compact. The processing system is divided into two blocks, PL(Programmable Logic) and PS(Processing system). PL is consisted of many parallel logic arrays and it can handle large amount of data fast, and it is designed in hardware-wise. PS is consisted of conventional processing unit like ARM processor in hardware-wise and sequential processing algorithm is installed on it. Consequentially HW/SW co-designed FPGA system is used for processing input images and measuring a relative 3D position of the leader, and this system showed RMSE accuracy of 0.42 cm ~ 0.51 cm.

Key words: Co-design, Formation Flight, Position Measurement, Vision based

1. Introduction

In recent years, UAVs have performed a various missions, such as surveillance, observation and information transmission. Formation flights are essential in order to increase the efficiency and success rate of these missions.

A relative position is very important and essential information to maintain formation during flight. To obtain this information, we can use the GPS based system, vision based system, RADAR and so on. Recently, vision based system on this subject has been on the rise because of robustness about GPS jamming problem and archiving communication free environment.[1-5] However, vision based system requires large amount of computational capability in order to process the image information maintaining high update rate.

As a similar research topic, aerial refueling and formation flight of UAVs using vision based algorithm was researched. Aerial refueling of UAVs using only image information [1] was researched, and five red LEDs and the coordinates of the feature points on the two-dimensional plane of the camera were used to determine relative position. In this paper, the Hungarian method as a point matching algorithm was used and maximum 5Hz update rate was shown when using Intel Pentium 4 1.5GHz CPU. Vision based formation flights of UAVs[2] was researched, and high-intensity LEDs and an EKF (Extended Kalman Filter) were used to determine relative position data of UAVs by combining GPS and IMU (Inertial Measurement Unit) data. The update rate was up to approximately 25 Hz using a Fit-PC2 based Linux operating system.

Previous researches showed 5Hz and 25Hz update rates, but

* Graduate Student
** Graduate Student
*** Professor, Corresponding author, baenggi@pusan.ac.kr

Received: October 8, 2015 Revised: May 8, 2016 Accepted: June 15, 2016

http://ijass.org pISSN: 2093-274x eISSN: 2093-2480

it is still not sufficient for high speed UAVs. Thus FPGA based processing system is used to increase update rate in this research. FPGA has difference with ordinary microprocessor. It is consisted of programmable logical gate elements and reconfigurable internal wires. Important advantage of FPGA is that user can design and implement algorithm directly in hardware-wise using HDL(Hardware description language) in the PL block which is consisted of various logical elements like AND, OR, NOT gates. Using the FPGA, we can process the data in parallel and input image can be processed faster than conventional sequentially processing microprocessors.

Conventional vision processing sequence is can be divided into two steps, image acquisition and processing. When process the image by only software, image processing should be conducted after image acquisition, thus unnecessary time delay have to be occurred. In contrast using the FPGA hardware, each pixel can be processed immediately after receiving pixel data. Thus required time of image pre-processing including image thresholding and labeling can be dramatically reduced like Fig.1.

Thus relative position measurement system based on FPGA was researched. Using this system, heavy vision processing algorithm can be processed fast in parallel without loss of measurement accuracy. Simulation using synthetic image was conducted to verify performance of this system, and ground experiment using real image was conducted to verify the feasibility.

In chapter 2, system configuration will be introduced. Measuring relative position algorithm will be introduced in hardware-wise and software-wise in chapter 3. And then simulation results and field experimental results are shown in chapter 4. Finally we will conclude our research in chapter 5.

2. System configuration

2.1 Composition of equipments

The composition of equipments is shown in following

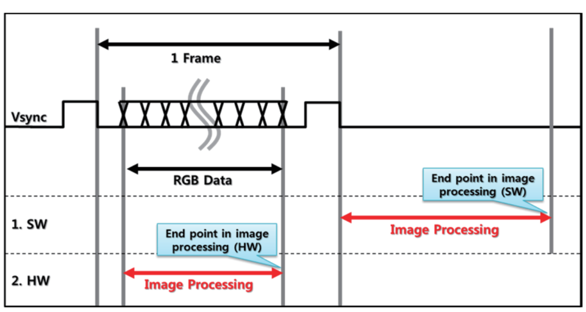

Fig. 1. Compare the software and hardware implementation in image processing

Fig.2. Virtex-5 xc5vsx95t FPGA chipset was used for the system. An embedded ADC (Analog to Digital Converter) circuit was used to convert input analog image data into digital data. An embedded DAC (Digital Analog Converter) circuit was used to convert image of extracted feature points and ROI (Region Of Interest) into analog signal to export image to a monitoring device. An UART (Universal Asynchronous Receiver / Transmitter) circuit was also used to collect the result data.

2.2 FPGA IP configuration

Total IP configuration of the system is in Fig. 3. Some part of PL can be utilized as a softcore using the IP named Microblaze provided from Xilinx Inc. The softcore has a 32-bit RISC(Reduced Instruction Set Computer) architecture and also support pipeline and floating-point unit. Each IP(Intellectual Property) is connected to a PLB(Peripheral Local Bus) used to send and receive data between the softcore and other part of PL.

In the other part of PL, peripheral IPs such as timer IP, UART IP, and PIC (Peripheral Interrupt Controller) IP were assigned and connected to the softcore. The timer IP was used to check whether it is possible to process 60 fps image inputs in real-time and the UART IP was used to collect the relative position coordinate data of the leader. The PIC IP was used to allow the interrupt function to be used in the softcore IP.

3. Relative position measurement

A relative position of the leader can be obtained using the following 7 steps shown in Fig.4.

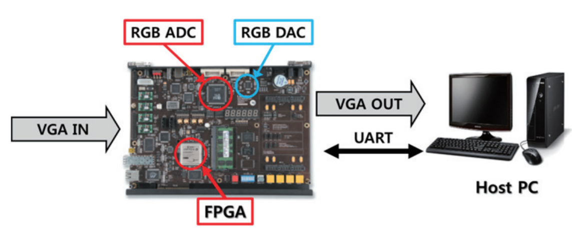

Fig. 2. Composition of equipments

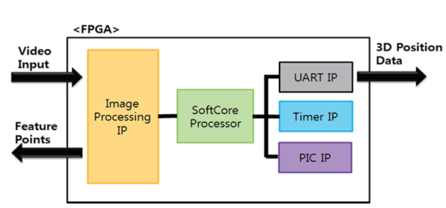

Fig. 3. IP configuration of FPGA

3.1 Hardware implementation on PL

3.1.1 Coordinate generator IP

VGA signals used in this research are composed of VCLK, VSYNC, HSYNC and DATA signals to complying VGA standards. VCLK serves to distinguish each pixel of the images, VSYNC informs start and end of a frame, HSYNC informs start and end of a horizontal line of an image, DATA inform RGB(red/green/blue) values of each pixel. Using these information, coordinate of current pixel was obtained.

3.1.2 Extraction of feature points

To extract red feature points from the RGB image, we converted RGB image into binary image using a threshold value obtained from locally adaptive thresholding algorithm in chapter 3.2.2.

Designed feature extraction IP was verified through functional simulation. The RGB data written continuously and in consecutive order as you can see in the Fig. 5. ([150, 20, 20], [160, 70, 70], [140, 40, 40]) In the first cycle in Fig. 5, threshold was set to [100, 50, 50] for each color plane, both the first and third feature point were extracted as a feature points, but in the second cycle, only the first data was extracted as a feature point because threshold was changed to [100,30,30] by the locally adaptive thresholding algorithm. Detailed feature extraction process is as below.

To extract the red feature points of 3D UAV model in Fig. 6, at first, pixels in red plane which have larger value than the given red threshold were extracted as shown in Fig. 7. Second, pixels in green and blue plane which have less value than the give green and blue threshold were extracted. And then AND gate logic was used for finding pixels that meets the three conditions as shown in Fig. 8.

3.1.3 Line buffer

Prior to performing the labeling algorithm, prior pixel information should be rearranged in a line buffer and we

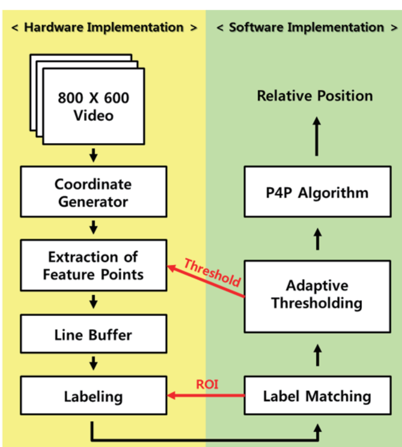

Fig. 4. Measurement flowchart of relative position of leader

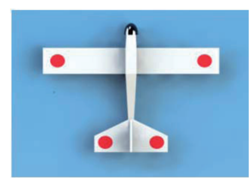

Fig. 6. 3D modeling of UAV

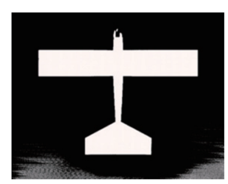

Fig. 7. Extraction of red plane

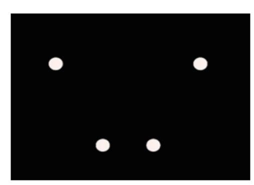

Fig. 8. Extracted red points

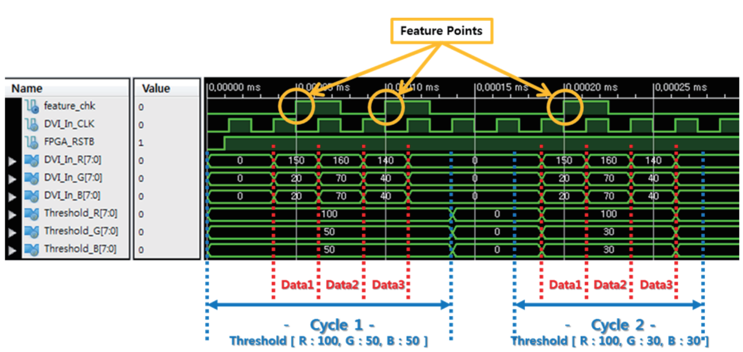

Fig. 5. Functional simulation of extraction of feature points

DOI: http://dx.doi.org/10.5139/IJASS.2016.17.2.232

used a register as a line buffer. All of the information from 1-1 to 3-3 in Fig. 9 is stored in line buffer. Considering overlooking camera installment angle, horizontal length of almost feature points was longer than vertical length in an image. Thus we chosen 7×5 sized mask similar aspect ratio of $\sqrt{2}$: 1.

3.1.4 Labeling

Labeling algorithm assigns label to the current pixel to distinguish current object between many objects. The input image pixel is received from the upper-left corner and if there's a meaningful pixel, then a new label is assigned to the pixel.

Assuming that pixel "IN" in Fig. 9 is extracted from the feature points, then the 7×5 mask is set and find any previous label in this mask. If there's other pixel which has a label, then current pixel is assigned not a new label but the existing same label as shown in Fig.10. If there is not any label in mask, new label is assigned to the pixel "IN" as shown in Fig.11. Fig. 12 shows a result image indicating each label using distinguishable different color. In addition, we set four ROI(Region of interest) to suppressing influence of noisy environment during the flight, and labeling was performed in these regions only.

1-1	1-2	1-3	1-4	1-5	1-6	1-7
2-1	2-2	2-3	2-4	2-5	2-6	2-7
3-1	3-2	3-3	IN			

Fig. 9. 7×5 Mask using line buffer

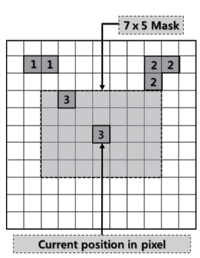

Fig. 10. Example of labeling 1

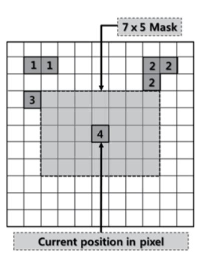

Fig. 11. Example of labeling 2

3.2 Software implementation on PS

3.2.1 Label matching

The label matching was designed to implement on the softcore IP and it has four functionalities. First, it can prevent possibility of choosing noise as a feature point like Fig. 13. Second, it maintains the number of label for the continuous frames during the change of number of label as shown in Fig. 14. Third, it calculates center of feature points. Finally it updates ROI continuously with respect to the center of each feature points.

3.2.2 Locally adaptive thresholding

After set the ROI around the feature point such as Fig. 15, average of pixel data $[R,G,B]_{se}$ and $[R,G,B]_{fp}$ is utilized as an updated threshold value following equation (1).

$$[R,G,B]_{th} = ([R,G,B]_{se} + [R,G,B]_{fp})/2 \quad (1)$$

3.2.3 Position measurement

By reconstructing the 2D coordinates of the feature points into 3D coordinates, it is possible measuring a

Fig. 12. Labeled feature points

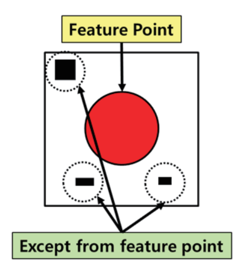

Fig. 13. Diagram of except feature point

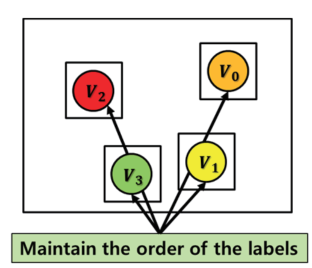

Fig. 14. Diagram of maintain the order of the labels

relative position of a leader with respect to a follower, and Perspective 4- Points (P4P) algorithm was used for the reconstruction. [9]

The position vector of the feature points marked on the UAV is defined as P_i (i=0,1,2,3). When we see P_i from the monocular vision, the position vector can be projected onto the image plane and it is defined as V_i (i=0,1,2,3), as shown in Fig. 16. We can find the value of α and β that satisfies the following equation from the known information.

$$P_3 = P_0 + \alpha(P_1 - P_0) + \beta(P_2 - P_0) \tag{2}$$

The relationship between P_i and V_i is defined as P_i=k_iV_i. When we substitute P_i=k_iV_i into the Eq. 2, following Eq. 3 can be obtained

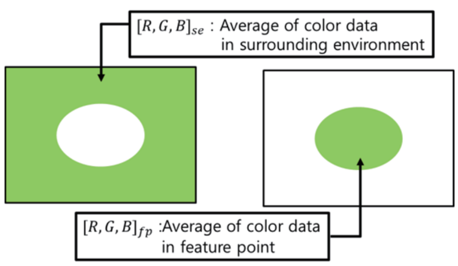

Fig. 15. Concept of locally adaptive thresholding

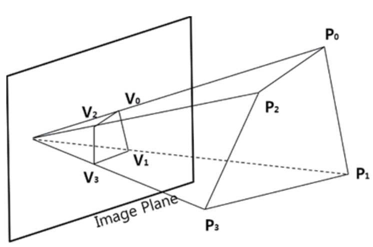

Fig. 16. Geometry relation between P_i and V_i

$$\frac{k_0}{k_3}(1 - \alpha - \beta)V_0 + \frac{k_1}{k_3}\alpha V_1 + \frac{k_2}{k_3}\beta V_2 = V_3 \tag{3}$$

The $[V_0,V_1,V_2]$ matrix is independent and can be inverted, thus value of $\frac{k_0}{k_3}, \frac{k_1}{k_3}, \frac{k_2}{k_3}$ can be obtained, from the above equation. Furthermore, the values of k_0,k_1,k_2 can be determined after obtaining the value of k_3 from Eq.4.

$$k_3 = \|P_0 - P_3\|_2 / \left\|\frac{k_0}{k_3}(1 - \alpha - \beta)V_0 - V_3\right\|_2 \tag{4}$$

As a result, it is possible to estimate the relative position of the leader in 3D coordinates using a feature points on the image in 2D coordinates. However, this data is a value in the camera coordinate system. To transform the coordinate system from camera to body frame of an UAV, a direction cosine matrix considering camera installment angle was used.

3.2.4 Low-pass filtering of relative position

The results of calculating the relative position in image has high frequency error, because the change of the number of pixel significant impact on the relative position measurement. Therefore, we designed the 1st low-pass filter.

$$\bar{x}_k = \alpha \cdot \bar{x}_{k-1} + (1 - \alpha) \cdot x_k \quad ,(0 < \alpha < 1) \tag{5}$$

4. Experimental results

4.1 Graphic User Interface

For convenience, GUI(Graphic User Interface) was made like Fig. 17. It is possible to store the results data with UART interface and uplink the ROI information to the FPGA system. Through the GUI, results from the FPGA system including

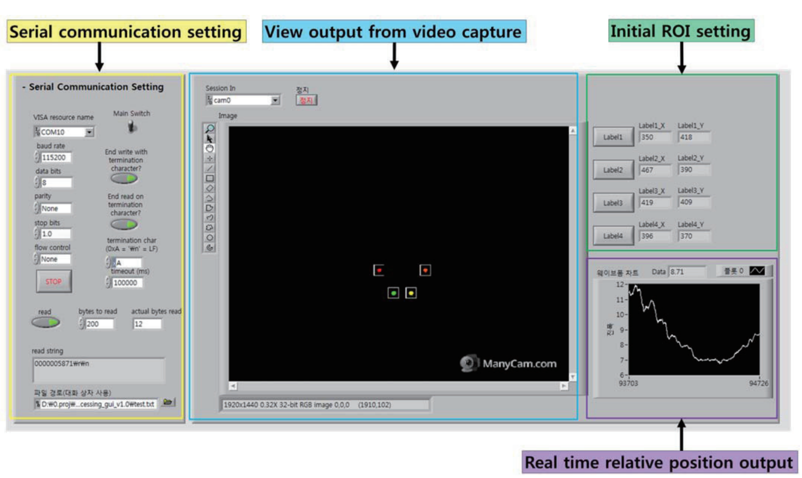

Fig. 17. Graphic user interface

results of the label and ROI set can be checked.

4.2 simulation test using synthetic image

4.2.1 Creation of flight video for simulation

In order to watch the four feature points on the leader's wings, the follower was positioned at back and upper position like Fig. 18. To track the target on the center of image, camera installment angle was set to look down in 45 degree.

To simulate formation flight situation, simulation video was created like Fig. 19. Leader aircraft do the planned maneuver like moving forth and back, right and left or up and down. Information of the maneuver was stored and this information was used to calculate position error.

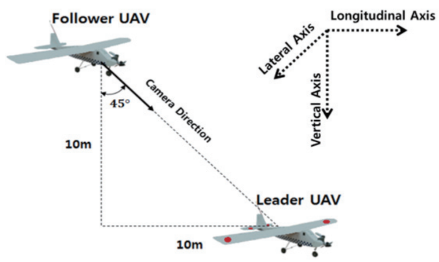

Fig. 18. Position of leader and follower

Fig. 19. Simulation image

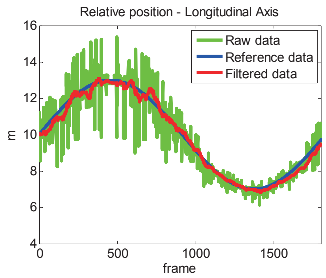

Fig. 20. Relative position on longitudinal axis

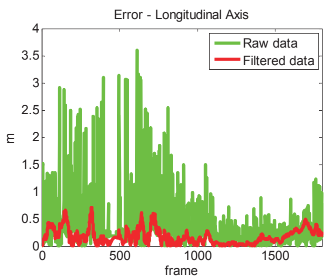

Fig. 21. Position error on longitudinal axis

4.2.2 Longitudinal movement

In this simulation, leader aircraft was moved forth and back up to 3m along the longitudinal axis. The 3D position was calculated using the FPGA system, and the error of the position was obtained as shown in Fig. 20 and Fig.21, and RMSE and maximum error was 0.0051m and 0.7136m each.

4.2.3 Lateral movement

In this simulation, the leader aircraft was moved right and left 3m each along lateral axis, and RMSE and maximum error was 0.0046m and 0.4492m each.

4.2.4 Vertical direction movement

In this simulation, the leader aircraft was moved up and down up to 3m each along vertical axis, and RMSE and maximum error was 0.0042m and 0.6377m each.

4.3 ground experiment using real image

In the simulation, feasibility of this algorithm is hard to be verified, thus ground experiment using real image was conducted. Four markers were attached on a UAV in Fig. 26. We installed camera by looking down on 45-degree where 10m apart in the vertical and longitudinal axis from the leader aircraft.

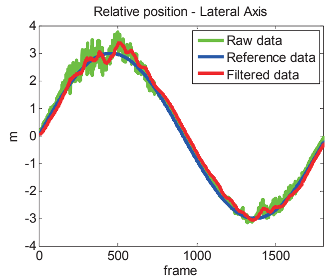

Fig. 22. Relative position on lateral axis

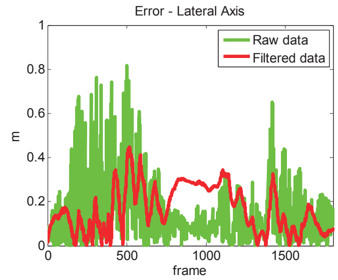

Fig. 23. Position error on lateral axis

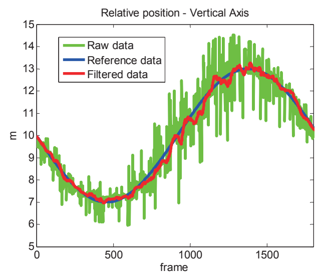

Fig. 24. Relative position on vertical axis

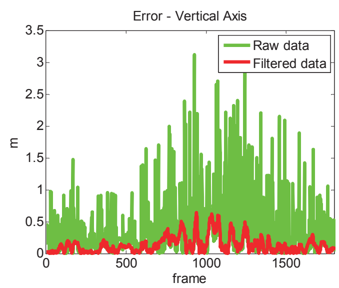

Fig. 25. Position error on vertical axis

Fig. 26. Real image for noise test

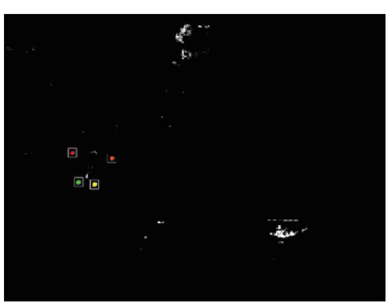

Fig. 27. Feature points and ROIs in real image

Table 1. Results of the data in all axes

	Longitudinal		Lateral		Vertical	
	Raw	Filtered	Raw	Filtered	Raw	Filtered
RMSE (m)	0.0135	0.0051	0.0049	0.0046	0.0120	0.0042
Max Error (m)	3.6045	0.7136	0.8167	0.4492	3.1177	0.6377

4.4 Result of ground experiment

4.4.1 Measured relative position

In this ground experiment, true position of the leader aircraft can't be obtained, thus we just observed performance of this system using the obtained relative positions. The results of the data were measured like Fig. 28, while moving the leader aircraft right to the left about up to 3m each.

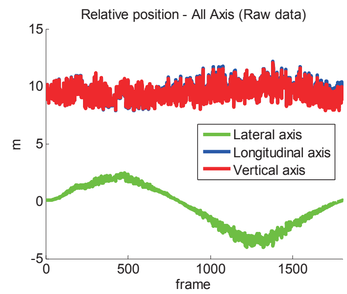

Fig. 28. Relative position in all axes(Raw data)

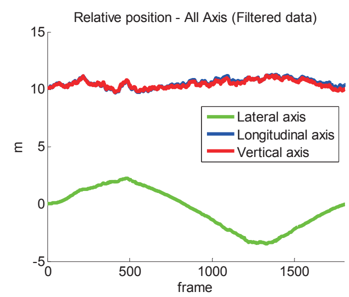

Fig. 29. Relative position in all axes(Low-pass filtered)

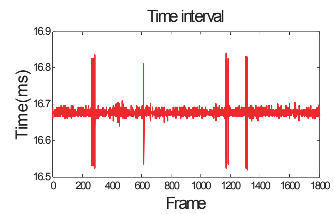

Fig. 30. Time interval in results data

Table 2. Results of the update rate data

Average(ms)	Measure(Hz)
16.67	59.96

DOI: http://dx.doi.org/10.5139/IJASS.2016.17.2.232

(232~239)15-158.indd 238 2016-06-30 오후 3:13:15

4.5. Update rate of the system

Update rate was measured using the timer IP and average update rate was about 60 Hz.

5. Conclusion

In this paper, FPGA based vision processing system to measure the relative position of the leader aircraft was shown. Several algorithms were divided into two groups according to the implementing location. In the PL, extracting feature points and labeling algorithm were implemented in hardware-wise. After handling large amount of data fast using the configured PL, sequential logical algorithms were implemented on the softcore named Microblaze with additional add-on IPs. By applying adaptive thresholding algorithm, we can obtain feature points more clearly. Also we can assign same label to a feature points for connected frames due to applied label matching algorithm.

The P4P algorithm implemented on the softcore IP was used to reconstruct a relative position of a leader. As a consequence, RMSE of measuring relative position was shown as 0.0051m along the longitudinal axis, 0.0046m along the lateral direction and 0.0042m in vertical direction. Performance of the lateral direction was better than other axis direction, and it is because uncertainty due to loss of depth information in 2D image affects longitudinal and vertical direction when camera installed to look down. Relatively lateral movement of the vehicle can more accurately measure due to uncertainty-free condition.

To verify the feasibility of this system, ground experiment was conducted, and position of the leader aircraft was obtained well. Due to adaptive thresholding algorithm, four feature points are tracked well and accurately.

Main contribution of this research is designing technique for embedded FPGA based system to conduct formation flight between UAVs, and verification of the feasibility in hardware-wise and software-wise. In the future, implementing additional filters like Kalman filter or particle filter increasing accuracy and robustness of the system will be required. Guidance and control law for the formation flight using flight simulators will be also required to verify the more feasibility.

Acknowledgement

This paper was supported as part of execution result of "A study on the design of integrated controller for actuation systems based on mixed signal FPGA" from Agency for Defense Development of Korea.

References

[1] Kim, Y. J., Lee, J. Y., Park, H. J., Lee, D. and Bang, H., "Real-time Implementation of Vision-based Pose Estimation for UAV Aerial Refueling" *KSAS Spring Symposium*, 2011, pp. 633-637.

[2] Oh, S. M. and Johnson, E. N., "Relative Motion Estimation for Vision-based Formation Flight using Unscented Kalman Filter", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2007.

DOI : 10.2514/6.2007-6866

[3] Mahboubi, Z., Kolter, Z., Wang, T., Bower, G. and Ng, A. Y., "Camera Based Localization for Autonomous UAV Formation Flight", *AIAA Infotech@Aerospace 2011*, Vol. 3, pp. 2631-2643.

DOI : 10.2514/6.2011-1658

[4] Kim, J. H., Heo, J. W., Shin, J. K., Lee, D. W., Cho, K. R. and Hur, G. B., "Measurement of the 3-dimensional Relative Position of the Leader during Formation Flight" *KSAS Fall Conference*, 2012.

[5] Cheon, B. K., Kim, J. H., Lee, D. H. and Cho, K. R., "Guidance and Control Algorithm Considering Position and Attitude of Leader for Formation Flight" *KSAS Fall Conference*, 2014, pp. 353-356.

[6] Elamaran, V., Praveen, A., Reddy, M. S., Aditya, L. V. and Suman, K., "FPGA Implementation of Spatial Image Filters using Xilinx System Generator", *ICMOC 2012*, Vol. 38, pp. 2244-2249.

DOI : 10.1016/j.proeng.2012.06.270

[7] Raut, N. P. and Gokhale, A. V., "FPGA Implementation for Image Processing Algorithms Using Xilinx System Generator", *IOSR-JVSP*, Vol. 2, 2013, pp. 26-36.

DOI : 10.9790/4200-0242636

[8] Moreo, A. T., Lorente, P. N., Valles, F. S., Muro, J. S. and Andres, C. F., "Experiences on Developing Computer Vision Hardware Algorithms using Xilinx System Generator", *ICMMM*, Vol. 29, 005, pp. 411-419.

DOI : 10.1016/j.micpro.2004.11.002

[9] Hung, Y., Yeh, P. S. and Harwood, D., "Passive Ranging to Know Planar Point Sets", *Proc. IEEE International Conference. Robotics and Automation*, 1985, pp. 80-85.

DOI : 10.1109/robot.1985.1087376