Applying FPGA to Implement Real-Time Infrared Tracking of Moving Objects

Jen-Yu Shieh, Yi-Lin Liao, Wei-Cheng Huang, Kun-Hsien Lin

Graduate Institute of Electro-Optical and Material Science,
National Formosa University, Huwei Township
Yunlin County 632, Taiwan

***Abstract*—Based on optical image analysis characteristics, various applications can be developed for infrared monitoring systems. Because infrared images are characterized by poor quality, low contrast, and few feature points, median filters are employed to recover and reconstruct images to improve the quality of infrared images. Based on the theory of order statistics, median filters are an easy-calculating and fast non-linear signal processing technology that can denoise signals effectively. After pre-processing, the reference background images are established and saved on SDRAM. We then retrieve the sequence images and the reference background images to conduct background subtraction to calculate image differences. Following background subtraction, changes in light sources and the speed of moving objects can cause noise and object appears shattered. Thus, we use morphological closing operation to mend shattered, incomplete segments and morphological opening operation to remove unnecessary noises and small segments. To realize moving object tracking, we employ image labeling to segment various moving objects and the barycentric method to calculate the barycentric and boundary coordinates. Finally, the moving objects were retrieved.**

***Keywords*: FPGA, median filter, background subtraction, image labeling, object tracking**

I. Introduction

Most monitoring systems are PC platform based and have the disadvantages of low efficiency and space-occupation. The system core of this study was an Altera Cyclone II FPGA DE2-70. Using the ADV7180 analog-to-digital conversion and the ITU-R 656 decoding circuit, interlaced scans were conducted on the images using SDRAM as the data buffer and numerous modules to calculate video algorithms. YCbCr signals were then converted to RGB color spaces. Finally, the processed results were displayed on an LCD. The system was completed by the Verilog hardware circuit that enables the rapid application of algorithms on images, thereby realizing real-time recognition.

II. Literature Review

A. Median filters

To improve the disadvantages of infrared images (e.g., poor quality, low contrast, few feature points, small grey-level difference, and boundary ambiguity), we employ median filters to remove image noise. Lu et al. [1] realized the optimal order algorithm median filter using a field-programmable gate array (FPGA), and the experimental results proved that the median filter can effectively filter grey-level images of 1K × 1K within approximately 20 ms, verifying the practical application of median filter systems. Yang et al. [2] also realized a median filter using a FPGA. Median filters are a non-linear signal processing technology based on the theory of order statistics that can denoise efficiently. A filter module is comprised of two FIFOs and seven registers. The filter module can filter image medians successfully using a 3 × 3 mask in a sample and fast operation, and can adequately protect the image detail when impulse noises occur.

B. Moving object tracking

The tracking of moving objects is the first step of object motion analysis. The objects' movements are retrieved by algorithms and analyzed by a sequence of temporal motion data. The object motion analysis is comprised of four procedures: video capturing, motion detection, motion tracking/behavior understanding, and object motion. The analysis methods include the motion energy method [3], optical flow method [4], and background subtraction [5, 6, 7].

1) Motion energy method

The motion energy method is also called the time difference method. Under the premise of static cameras and background, if the position of an object in the frame changes, the background of the object from the preceding frame is revealed and the position of the current object would cover the background of the preceding frame. This phenomenon can cause significant image changes in brightness. By detecting changes of brightness in sequence images and filtering noises at an appropriate threshold, we can identify the dynamic and static regions of images, as presented in Eq. (1):

$$M_n(u,v) = \left\{ \begin{array}{ll} 1, & if \left| I_n(u,v) - I_{n-1}(u,v) \right| \geq \varepsilon \\ 0, & otherwise \end{array} \right\} \tag{1.}$$

where $I_n(u,v)$ and $I_{n-1}(u,v)$ are the brightness values of a given position of two images, ε is the threshold, and $M_n(u,v)$ is the binary image subtracted by any two-image sequences. This method is simple; however, when the movement of targets is too large, the method cannot calculate the post-movement position precisely, resulting in errors. To resolve this problem, temporal and spatial gradients are integrated to obtain the moving boundary of the targets. The temporal gradient is the motion energy method, and the spatial gradient marginalizes current images to obtain image boundaries. Following spatial and temporal gradient operations, the boundary of moving targets is calculated, as presented in Eq. (2).

$$ME(u,v) = \begin{cases} 1, & if\ M_n(u,v) = 1\ and\ E_n(u,v) = 1 \\ 0, & otherwise \end{cases} \qquad (2.)$$

where M_n is the motion energy method in Eq. (1), and E_n is the image marginalized from the current image.

2) Optical flow method

Optical flow is the speed of a point in an image moving from direction u to direction v or the flow direction of lights. Previous studies have used optical flow for high-precision 3D reconstruction [8] or activity analysis [9]. The advantage of the optical flow is its ability to separate different types of motion in an image. If an image shows background motion caused by camera and object motion, the optical flows presented are different. Previous studies on the optical flow method have hypothesized that the image brightness of targets at time t is equivalent to that at time t+Δt (Δt is small between two sequence images). In other words, the image brightness of the target at the same point to the time differential is zero. Therefore, if I(u,v,t) denotes the brightness of point (u,v) on a flat image at time t, the assumptions above can be expressed as Eq. (3).

$$\frac{dI(u,v,t)}{dt} = 0 \qquad (3.)$$

If Eq. (3) is extended by the chain rule (assuming I(u,v,t) are differentiable), Eq. (3) can be revised as Eq. (4):

$$\frac{\partial I(u,v,t)}{\partial u}\frac{\partial u}{\partial t} + \frac{\partial I(u,v,t)}{\partial v}\frac{\partial v}{\partial t} + \frac{\partial I(u,v,t)}{\partial t}\frac{\partial t}{\partial t} = 0 \qquad (4.)$$

Because $\partial t / \partial t = 1$, $\partial u / \partial t = u_t$ and $\partial v / \partial t = v_t$, Eq. (4) can be revised as Eq. (5):

$$I_u u_t + I_v v_t + I_t = 0 \qquad (5.)$$

where I_u, I_v, and I_t present the partial differentiations of direction u, direction v, and time t, respectively. Therefore, Eq. (5) is called the brightness limit equation, where $\partial u / \partial t = u_t$ and $\partial v / \partial t = v_t$ represent the respective moving speeds of the target in directions u and v, that is, optical flows. In addition to Eq. (5), other limits must be added to calculate these two values. Therefore, numerous scholars have proposed various methods and added limits to calculate the optical flows of images. Barron et al. [10] organized and classified four primary methods: differential, region-base matching, energy-base, and phase-base methods. However, the data size of images is very large (e.g., a 640 × 480 p greyscale image at has a total of 307200 pixels), and the calculation of optical flows is complicated [11]. Calculating the optical flow of each pixel in an image is time consuming. Thus, without the assistance of special hardware, the optical flow method is inadequate to apply for real-time calculation.

3) Background subtraction

Before the system conducts detection, a reference background model of the detection region must be established in advance. If any unexpected object intrudes into the model, the object would cover the original background, resulting in changes in brightness. Therefore, by detecting the brightness difference between the current image and background model and filtering noises at a specific threshold, we can immediately calculate the position of the moving object precisely, as presented in Eq. (6):

$$b_n(u,v) = \begin{cases} 1, & if \left| I_n(u,v) - B_n(u,v) \right| \geq \varepsilon \\ 0, & otherwise \end{cases} \qquad (6.)$$

where I_n and B_n are respectively the current image and reference background model, ε is a threshold, and b_n is the binary image obtained by the absolute threshold of the brightness difference between the two images (i.e., the current image and reference background model), that is, the region of the moving object. Table 1 shows a comparison of the three primary motion detection algorithms.

Table 1. Comparison of motion detection algorithms

Algorithm/efficiency	Calculation volume	Detection quality
Optical flow	Large	High
Motion energy	Small	Low
Background subtraction	Small	High

Zafar et al. [12] suggested applying infrared tracking to improve military monitoring safety. Infrared monitoring systems that ensure military safety have become an essential night monitoring system. The system uses the background subtraction method to identify human objects to assist soldiers and commanders to make correct decisions, thereby enhancing military safety. Xiong et al. [13] proposed a background subtraction and frame differential integrated object detection method. Dhome et al. [14] showed that the application of the background subtraction method could obtain effective results.

III. THE SYSTEM ARCHITECTURE OF FPGA

C. Hardware architecture

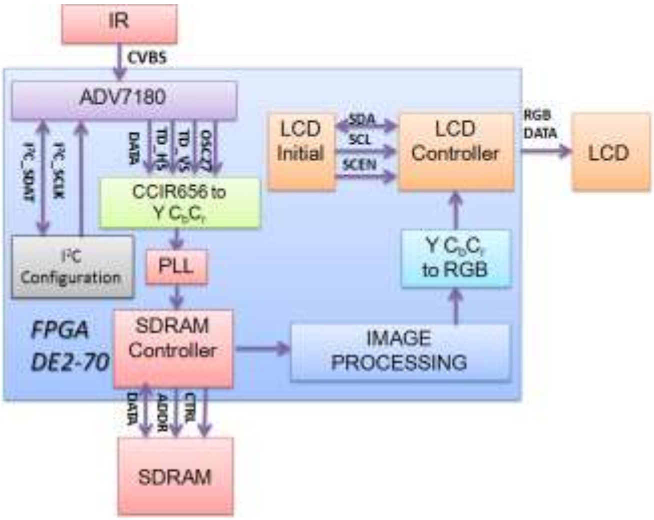


Figure 1. Hardware architecture

Infrared cameras send messages to the FPGA through the AV terminal. After a series of digital signal processes, the messages are displayed on LCDs. Through the ADV7180 analog-to-digital conversion and the ITU-R 656d decoding circuit, the images use the SDRAM as the data buffer and for conducting interlaced scans. After implementing the image algorithm through numerous modules, YCbCr signals were converted into RGB color spaces. Finally, the results were displayed on the LCD, as shown in Fig. 1.

The DE2-70 uses two SDRAM chips; one is for image frame buffering, and the other is for processing image algorithms. The SDRAM controller has four ports for saving and retrieving, and is capable of reading and writing simultaneously. The design procedure was to improve the system tick from 50MHz to 166MHz through the internal phase-locked loop (PLL) circuit of the FPGA to provide the operating frequency to the controller and the external SDRAM. The primary architecture of the SDRAM controller was composed of a command module, control interface module, and data path module, as shown in Fig. 2.

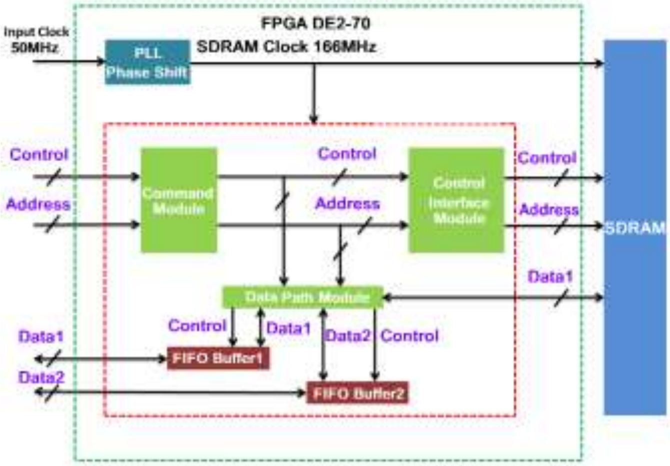


Figure 2. The hardware architecture of the SDRAM controller

D. Digital singal processing

Signals were sent to the TV decoder for analog-to-digital conversion, and then the FVH synchronizing signals were retrieved. Fig. 3 shows the horizontal signal wave composed of the start of active video (SAV) data, the end of active video (EAV) data, horizontal blanking (H Blank), and active video (Active Video). SAV and EAV were used to judge the active video input, which were composed of FF00XY; of which, XY was composed by 8 Bit data, as shown in Table 2.

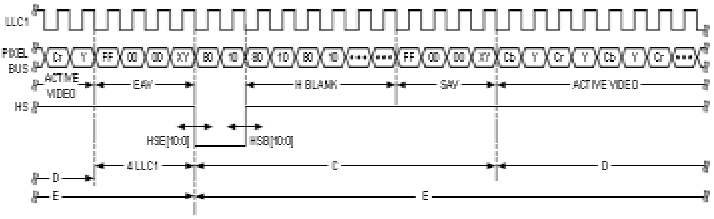


Figure 3. Scan line temporal graph

TABLE 2. THE COMPOSITION OF XY

Bit 0~ 3	Protection Bits According to ITU-656
Bit 4	H = 0 (SAV), and H = 1 (EAV)
Bit 5	V = 0 (Active Video), and V = 1 (Vertical Blanking)
Bit 6	F = 0 (Odd Field), and F = 1 (Even Field)
Bit 7	1

A line of video signals in streaming signals had 1440 entries of data, including the brightness signal Y and the chromas Cb and Cr. The form of 4:2:2 means that the length of an image was 720 pixels; of which, each pixel had a brightness signal, and each two pixel had chroma signals. Therefore, Y had 720 entries of data, and Cb and Cr had 360 entries of data each. The image signal sequence was Cb, Y, Cr, Y, Cb, Y, …, and so on. After the input of an image, Bit 4 judged whether there is a scan line entry. If Bit 4 = 0, it is interpreted as the start of the video data, and Cont begins counting. While Cont < 1440, Cont accumulates and Bit 4 is treated as an active video output value. If Cont > 1440, the system treats the Bit 4 value as

EAV, and TV_Y+1 re-detects the value of Bit 4. If Bit 4 = 1, it is interpreted as the end of the video data, and TV_Y+1 re-detects horizontal scan signals, as shown in Fig. 4.

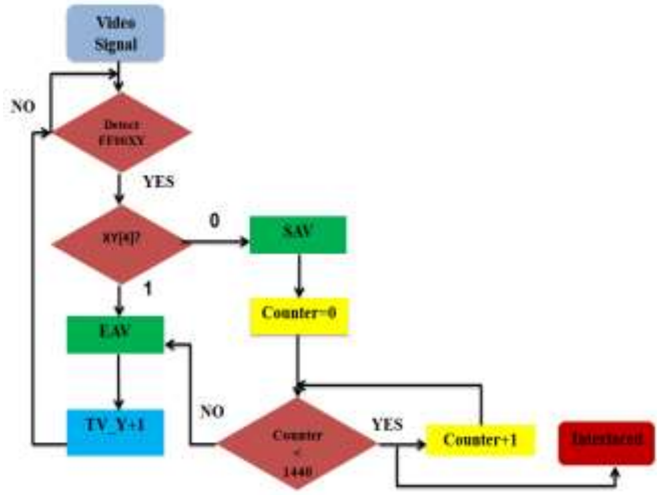


Figure 4. FPGA signal detection block graph

IV. Research Methods for Dynamic Tracking

This section details the image algorithms and circuit design methodology, as well as the hardware architecture discussed in Section 3. Fig. 5 shows the image processing procedure. The following procedures are performed in different modules.

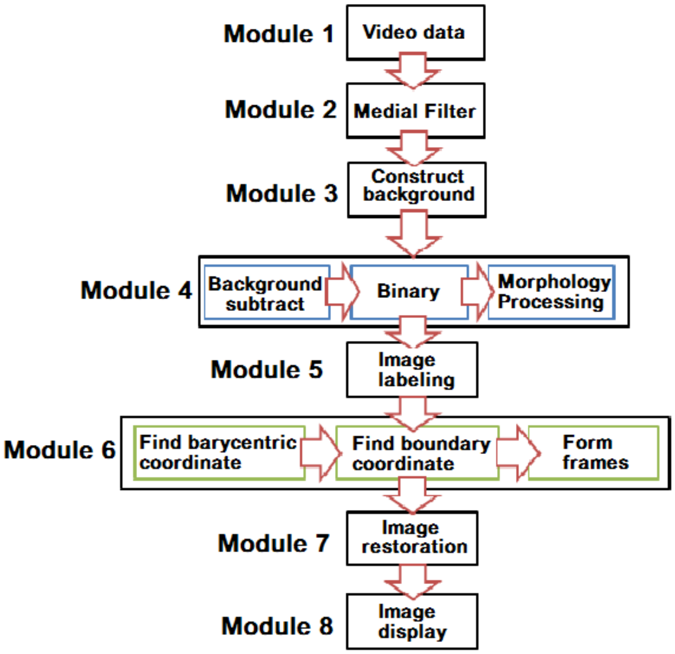
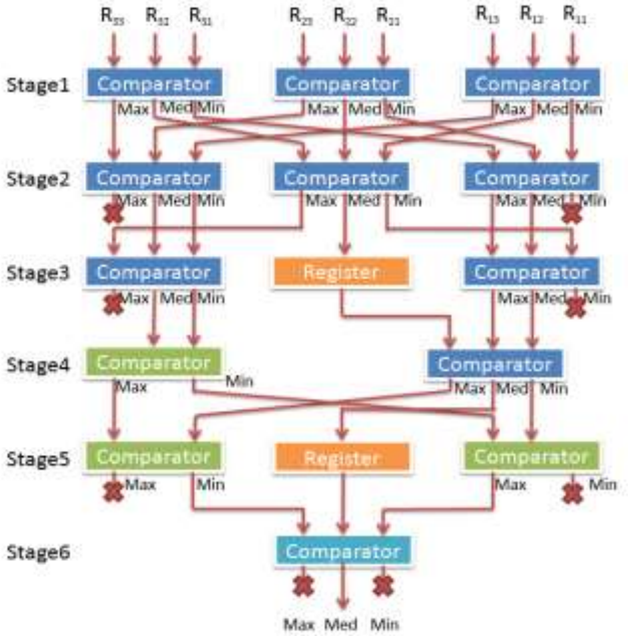


Figure 5. Image processing procedure

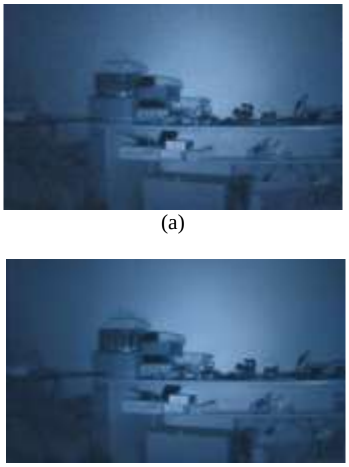
E. *Median filters*

Sorting algorithms are the core of median filters. The characteristics of horizontal processing of pure FPGA hardware design enhance the processing speed. It is difficult to identify the median of nine pixels; however, identifying the maximum and minimum is rapid. First, we identified the maximum and minimum of the nine pixels and eliminated the two vales. Then, we identified the maximum and minimum of the seven pixels. By repeating the process, the remaining pixel was identified as the median. Fig. 6 shows the system architecture of a median filter.

Figure 6. The system architecture of a median filter



We designed three 3-input comparators to realize the circuit. By dividing the nine pixels into three groups, we compared their values to identify the maximum, minimum, and median of each group. Then, we classified the results using the comparators into the maximum, minimum, and median groups and conducted the second-phase comparison. In the second phase, we identified the maximum and minimum among the nine values and eliminated the two values. In the third-phase comparison, we identified the maximum and minimum among the seven values and eliminated the two values. Then, we identified the maximum and minimum among the five values in the fourth and fifth comparison phases. Finally, we used a comparator to identify the median in the sixth phase. Additionally, to ensure the data synchronization in the calculation process, we added a register in the third and fifth phases. Therefore, we require ten 3-input comparators and two 2-input comparators. The comparison is shown in Fig. 7.



(a)

(b)

Figure 7. (a) The image with noises (b) The image after a median filter

F. Background subtraction

Figs. 8 and 9 show that the background image subtraction circuit was processed by two modules; one module established the background image circuit architecture and the other module established the image subtraction circuit architecture.

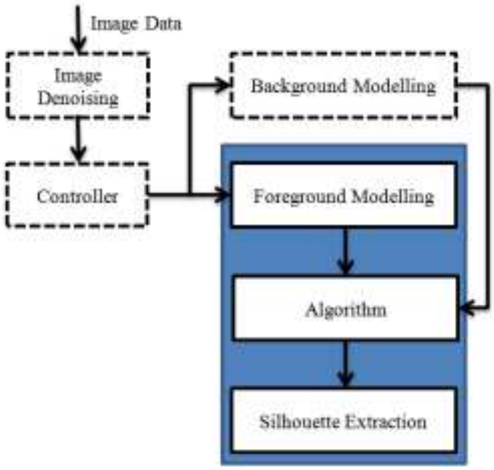


Figure 8. Background subtraction procedure

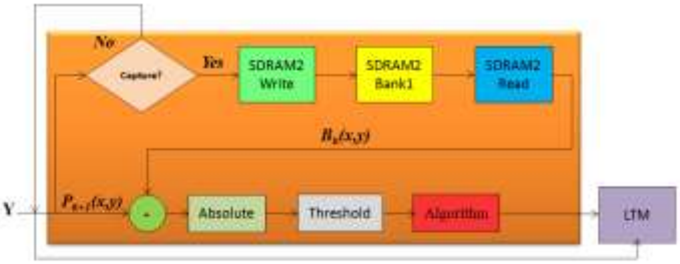
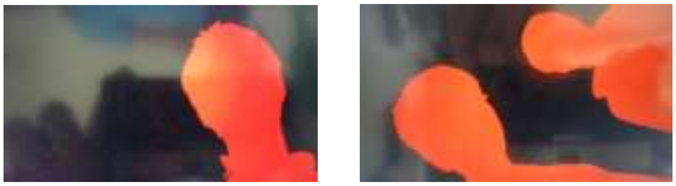


Figure 9. The hardware architecture of background subtraction

1) Establish background images through the Capture circuit and complete a new background image by writing the RAM Bank 1 of SDRAM2 through the internal SDRAM controller module of the FPGA.
2) Shoot a new image using the infrared camera and subtract it from the background image written on Bank 1 of SDRAM2.
3) The value after the subtraction is calculated and outputted through the absolute (ABS) circuit.
4) The output value is input into the threshold circuit. If the input image value is greater than the threshold, the image is eliminated and the output image value is set at zero. If the input value is greater than the threshold, the pixel is identified as a moving object and the output image value is set at 1023.
5) Following background subtraction, the image is calculated by dilation/erosion, binary, and image segmentation algorithms. Fig. 10 is the result of the background subtraction.



(a) (b)

Figure 10. Background subtraction (a) A single object (b) Two objects

G. Object tracking and detection

1) Forecast the position of the object after moving

First, we forecast the moving region [15] of the object and used n-1, n, and n+1 to present the adjacent images with the time interval Δt. In the n+1 image, the forecast position $\hat{P}_{n+1}$ of the moving object can be expressed by Eq. (7), as shown in Fig. 11.

$$\hat{P}_{n+1} = P_n + \vec{V}_n \Delta t \tag{7.}$$

Where, $\hat{p}_{n+1}$ is the forecast position in the n+1 image, P_n is the measured position in the n image, and $\vec{V}_n$ is the moving vector calculated by the n-1 and n images.

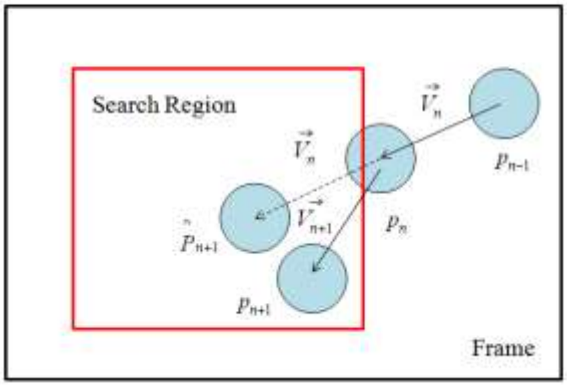


Figure 11. Forecast of the post-move position $\hat{p}_{n+1}$

Then, we define the search region (S) of the subsequent image (i.e., the possible region where the object move to), as expressed in Eq. (8).

$$S_{\hat{P}_{n+1}} = 2(2W_{P_n} + |V_n|) \times 2(2H_{P_n} + |V_n|) \tag{8.}$$

where W_{P_n} is the width of the object that uses P_n as the center in the *n* image, H_{P_n} is the height of the object that uses P_n as the center in the *n* image, and $S_{\hat{p}_{n+1}}$ is the search region that uses $\hat{p}_{n+1}$ as the center that enlarges upward and downward $2(2H_{P_n}+|V_n|)$ and left and right $2(2W_{P_n}+|V_n|)$.

2) The object comparsion before and after moving

In the two adajcent images, identifying the real position of the related object after moving is the objective of object tracking. We can identify the real position of the object by comparison. First, we must know the position of where the object identified in the n image is in the n+1 image to enable the comparison of the region (R) of the object in the n image with the R in the n+1 image. This method is called full search, as expressed in Eq. (9). Following calculation, $\hat{d} = \min C(d)$ is determined to be the optimal related position, that is, the real position of the object in the n+1 image. This is called the measure position.

$$C(d) = \sum_{x \in R} | I_n(x) - I_{n+1}(x+d) | \qquad (9.)$$

where $I_n(x)$ is the brightness on the position *x* in the *n* image.

Large objects or objects that move rapidly expand the object search region. Because full search requires abundant calculations, we can use moving directions, position forecast, and the size of objects to determine the position of moving objects directly. As expressed in Eq. (10), the reliable value D_n^{iv} (i.e., $\hat{d} = \max D_n^{iv}$) can be calculated to identify the optimal related position of the moving object.

$$D_n^{iv} = \alpha \left(\frac{V_n^v \bullet V_{n+1}^{iv}}{2|V_n^v||V_{n+1}^{iv}|} \right) + \beta \left(1 - \frac{|P_{n+1}^p - P_{n+!}^i|}{S_{n+1}^{MaxWide}} \right) + \gamma \left(\frac{2\sqrt{A_n^v \times A_{n+1}^i}}{A_n^v + A_{n+1}^i} \right) ,\alpha + \beta + \gamma = 1 \qquad (10.)$$

where V is the moving vector, P is the central position, and A is the area.

3) Tracking and updating decisions

Table 3 shows the five tracking situations. To make decisions according to specific situations, we can track covered objects and integrated moving objects using the system.

Table 3. Five tracking situations

Situation	Method
An existing moving object is not located in the search region.	The final searching procedure would locate the new objects.
Cannot find the moving object in the search region.	Declare the object as missing.
Find an object on the boundary region.	The optimal situation.
Find multiple objects in the search region.	Judge by Eq. (10)
Find objects in multiple search regions.	Full search

4) Searching for new objects or background updating

Searching for new objects is the final procedure of object tracking. The purpose of this procedure is to examine whether other objects or new objects that change from background to moving objects enter the frame. This procedure is similar to the background subtraction method. However, this procedure deletes the region that existing objects are located in from the image and the object regions related to the background prior to conducting background subtraction. This procedure focuses on the remaining objects in the image.

5) Barycentric coordinates

To retrieve moving objects, we must use a target coordinate as the basis of object tracking. The most rapid method is to identify the barycentric position of moving objects. First, we calculate the barycentric position of the x and y coordinates. The labled x and y coordinates of pixels and the product of the greyscale values of each pixel are summed. The result is divided by the total of labled pixels calculated by a counter. Then, the barycentric position of labled image is obtained and used to calculate boundary coordinates, thereby retrieving the moving objects, as shown in Fig. 12.

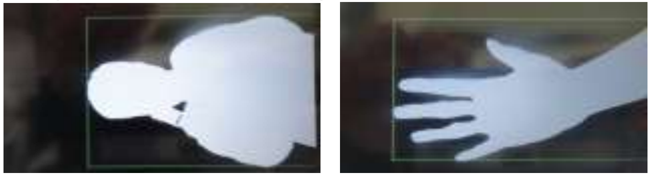


Figure 12. Moving object retrieval

6) Color space conversion

The digital signals output by the ITU-R 656 video decoding circuit are the brightness Y of 8 Bit and the chroma signals Cb and Cr. The LCD is shown in RGB format. Therefore, through the color space conversion equation (as expressed in Eq. (11)), the color space of YCbCr is converted into RGB color space.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 1.596 & 0 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y & -16 \\ C_b & -128 \\ C_r & -128 \end{bmatrix} \qquad (11.)$$

H. Pipeline processing

Pipeline processing [16] can increase time ticks, thereby enhancing implementation speed. However, the use of pipelines can increase the number of registers. For the FPGA, the use of pipelines is to increase logic elements; for the application-specific integrated circuit (ASIC), it is to increase areas, that is, costs. Therefore, we must use pipeline processing appropriately. Fig. 13 is the real-time signals from the FPGA generated by the logic analyzer. The algorithms were processed by the pipeline processing method. We saved the values of each calculation procedure in the register. Following this, the circuit was available for processing the next entry of data. Instead of processing the next entry of data after completing all the algorithms, this method prevents circuits from idling.

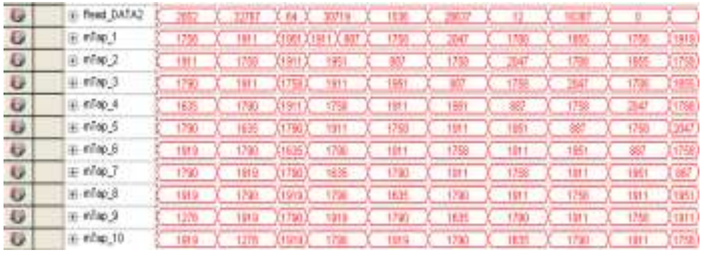


Figure 13. Logic analyzer signals

V. Experimental Results

The experimental results of this study are classified into two models that allow users to choose from. The difference of Models 1 and 2 is the display of the non-moving target (i.e., the background image). In Model 1, we tracked the outlines of the moving targets, and the background image was displayed. In Model 2, we tracked the moving objects, and the other non-moving objects were set as black pixels. Both of the models can improve the identification ability of monitoring systems.

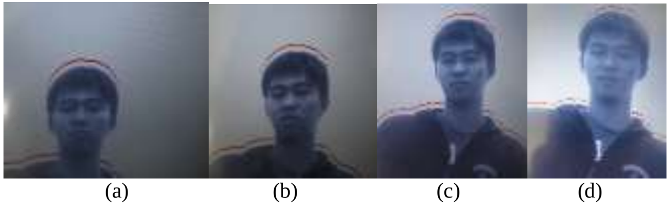


Figure 14. Images of single-person tracking frames at time (a) t_1, (b) t_3, (c) t_5, and (d) t_7.

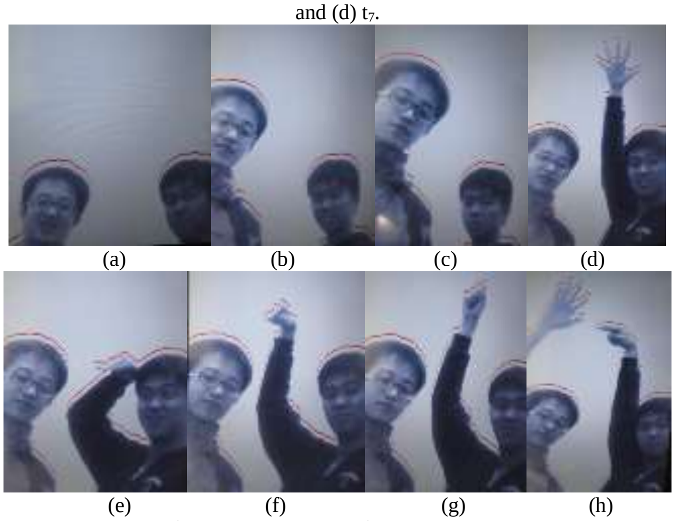


Figure 15. Images of two-people tracking frames at time (a) t_1, (b) t_3, (c) t_5, (d) t_7, (e) t_9, (f) t_{11}, (g) t_{13},and (h) t_{15}.

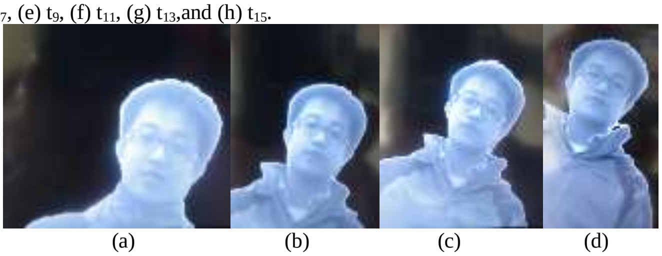


Figure 16. Images of single-person tracking frames at time (a) t_1, (b) t_3, (c) t_5, and (d) t_7.

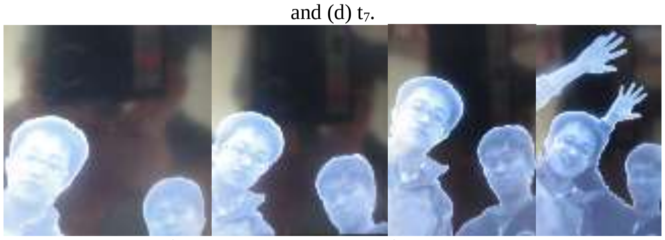


Figure 17. Images of the two-people tracking frames at time (a) t_1, (b) t_3, (c) t_5, and (d) t_7.

VI. Conclusions

This study used a FPGA to realize the real-time infrared tracking system for moving objects to enable traditional monitoring systems to develop the advantages of minimization and portability. The systems were designed using the Verilog hardware language because of its flexibility for software program design, the speed of hardware parallel processing, and the characteristics of anti-interference and function stability. The disadvantages of infrared images (e.g., poor quality, low contrast, few feature points, small grey-level difference, and boundary ambiguity) can cause identification difficulties for monitoring systems. Therefore, images must enhance quality and features through pre-processing. To denoise and improve image quality, median filters are an appropriate non-linear filtering technology. For hardware circuits, calculations for median filters are easy and fast. To track moving objects, we must establish reference background images and retrieve sequence images and reference background images to conduct the subtraction procedures to calculate differences in image values. Following background subtraction, changes in lighting and the speed of moving objects can cause noise and objects appear to be shattered. Therefore, we can use the closing operation of morphology to mend shattered, incomplete segments and the opening operation to remove noise and small segments. To process object tracking, we used image labeling to conduct the image segmentation of various moving objects. Then, we used the barycentric method to identify the barycentric and boundary coordinates. Finally, we retrieved the moving objects to realize the function of real-time tracking.

References

[1]. Lu Yan, Dai Ming, Jiang Lei, Li Shi, 2010, "Sort Optimization Algorithm of Median Filtering Based on FPGA", Machine Vision and Human-Machine Interface , pp. 24-25.

[2]. Yang Shao-yan, Liu Wan-cheng, Zhou Chang-chun, 2011, "FPGA Implementation for Improved Median Filtering Algorithm Based on IRFPA", Electro-Optic Technology Application, vol. 26.

[3]. Young-Kee Jung, Kyu-Won Lee, Yo-Sung Ho, 2004,"Content-based event retrieval using semantic scene interpretation for automated traffic surveillance", Intelligent Transportation Systems, vol. 2, No.3.

[4]. Simon Denman, Vinod Chandran, and Sridha Sridharan, 2007, "An adaptive optical flow technique for person tracking systems," Pattern Recognition Letters, vol. 28, no. 10, pp. 1232–1239.

[5]. Ahmed Elgammal,et al. 2002, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", vol. 90, pp.1151–1163.

[6]. Yu-Ting Chen, Chu-Song Chen, Chun-Rong Huang and Yi-Ping Hung, 2007, "Efficient hierarchical method for background subtraction", Pattern Recognition, vol. 40, pp. 2706– 2715.

[7]. Aguilar-Ponce, et al. 2005, "Real-time VLSI architecture for detection of moving object using Wronskian determinant", Circuits and Systems, pp. 7-10.

[8]. Matthew Brand, 2001, "Morphable 3D models from video", Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 456-463.

[9]. Alan Lipton, 1999, "Local application of optic to analyze rigid versus non-rigid motion", Technical Report, CMU-RI-TR-99-13, Robotics Institute, Carnegie Mellon University.

[10]. John Barron,David J. Fleet, Suzanne Beauchemin ,et al. 1992, "Performance of Optical Flow Techniques", Computer Society Conference on Computer Vision and Pattern Recognition , pp. 236-242.

[11]. Hai-Yan Zhang, 2010, "Multiple moving objects detection and tracking based on optical flow in polar-log images", Machine Learning and Cybernetics, vol. 3, pp. 11-14.

[12]. Iffat Zafar, et al. 2010, "Human silhouette extraction on FPGAs for infrared night vision military surveillance", Circuits,Communications and System , vol. 1, pp. 1-2.

[13]. Weihua Xiong, Junbin Guan, and Haipeng Pan, 2010, "A New Algorithm for Moving Objects Detection Based on Background and Consecutive Frames Subtraction", Intelligent Control and Automation , pp. 7-9.

[14]. Dhome Yoann, et al. 2010, "A Benchmark for Background Subtraction Algorithms in Monocular Vision: a Comparative Study", Image Processing Theory Tools and Applications, pp. 7-10.

[15]. Shyue- Wen Yang, et al. 2005, "VLSI Architecture Design for a Fast Parallel Label Assignment in Binary Image", International Symposium Circuits and Systems, vol. 3, pp. 2393–2396.

[16]. Devlin, Benjamin, et al. 2009, "Throughput Optimization by Pipeline Alignment of a Self Synchronous FPGA", Field-Programmable Technology, pp. 312-315.

AUTHORS INFORMATION

Jen-Yu Shieh received the B.S. degree in electrical enginerring from Feng-Chia University, Taichung, Taiwan, in 1984 and the M.E. degree in computer engineering and Ph.D. degree in electrical engineering from Florida Atlantic University, Boca Raton, in 1989 and 1992, respectively. While completing graduate degrees at FAU, he was an associate professor of Department of Electro-Optics Engineering at National Formosa University, Huwei, Taiwan. Since February 2010, he has been an professor. His current research interests include innovation and consumer products. He also has 47 patents and 4 consumer products.

Yi-Lin Liao was born in Taipei, Taiwan. He received the degree in Optical-Electro from the University of Taiwan, in 2009. He is a student of graduate institute of Electro-Optical and Materials Science, National Formosa University of Huwei, Yun Lin, Taiwan.

Kun Hsien Lin was born in Yun Lin, Taiwan, on March 28, 1988. He received the degree in Optical-Electro from the University of Taiwan, in 2010. He is a student of graduate institute of Electro-Optical and Materials Science, National Formosa University of Huwei, Yun Lin, Taiwan. He is engaged in researches on Electro-Optical and Materials Science. Mr. Lin is a member of National Formosa University

Wei-Cheng Huang was born in Changhua, Taiwan, on June 29, 1988. He received the degree in Optical-Electro from the University of Taiwan, in 2010. He is a student of graduate institute of Electro-Optical and Materials Science, National Formosa University of Huwei, Yun Lin, Taiwan.