Fast and Simple Motion Tracking Unit with Motion Estimation

Hyeon-cheol YANG[1], Yoon-sup KIM[1], Seong-soo LEE[1], Sang-keun OH[2], Sung-hwa KIM[2], Doo-won CHOI[2]
(1. *School of Electronic Engineering*, *Soongsil University*, *Seoul* 156-743, *Korea*;
2. *HyVision System Inc.*, *Seongnam* 462120, *Korea*)

Abstract – Surveillance system using active tracking camera has no distance limitation of surveillance range compared to supersonic or sound sensors. However, complex motion tracking algorithm requires huge amount of computation, and it often requires expensive DSPs or embedded processors. This paper proposes a novel motion tracking unit based on different image for fast and simple motion tracking. It uses configuration factor to avoid noise and inaccuracy. It reduces the required computation significantly, so as to be implemented on Field Programmable Gate Array (FPGAs) instead of expensive Digital Signal Processing (DSPs). It also performs calculation for motion estimation in video compression, so it can be easily combined with surveillance system with video recording functionality based on video compression. The proposed motion tracking system implemented on Xilinx Vertex-4 FPGA can process 48 frames per second, and operating frequency of motion tracking unit is 100 MHz.

Key words – active tracking camera; Surveillance system; differential image; moving picture encoding

Manuscript Number: 1674-8042(2010)03-0233-05

dio: 10.3969/j.issn.1674-8042.2010.03.07

1 Introduction

Surveillance systems are wide spread for crime prevention or disaster prevention. The system consists of several elements. Between them, an observation camera is one of the key elements. It has many advantages against supersonic or sound sensors, since they have distance limitation and little resistance to noise. Observation cameras are classified into by fixed camera, passive cameras with preprogrammed movement, and active cameras with adaptive motion tracking. Most conventional observation cameras are fixed or passive, but they often required camera to cover whole surveillance area. Active cameras with pan-tilt unit reacting movement of objects have more wide observation area, but they are often quite expensive since they require DSP or embedded processors. Moreover, Surveillance systems recently developed require that keep their data for analysis. The video data are very large to storing in observation camera; therefore it is necessary to reduce data and to select data.

This paper proposed fast and simple motion tracking unit using differential images. It is cost-effective and shows high surveillance performance without DSP or embedded processor. Furthermore, it also performs some calculation for motion estimation in video compression, so it can be easily combined with surveillance system with video recording functionality based on video compression.

2 Moving object detection methods

2.1 Using stored background image

At first, this method[1] stores for background image covering all possible camera's route. The stored image is like a panorama picture. The system finds background area in stored picture, and compares with stored picture, and then object is selected from different area between stored image and input image. This method is not related to velocity or specification of object. However, it suffers from large memory to store all background area. Especially, if background is changed, this method cannot find objects effectively. Therefore, it needs frequent refreshment of background image.

2.2 Using object features

Many objects have characteristic features such as color or shape[2]. The system makes model using these features and find best similar area from input image as the object. It has not only simple calculation but also easies to find object when features of object is easily distinguishable. However, it requires accurate model to find objects. Furthermore, it is difficult to recognize initial movement of object, and it is weak against light source variation or shape variation.

※ **Received:** 2010-06-29
Project supported: This work was sponsored by the MKE (The Ministry of Knowledge Economy, Korea), the ITRC (Information Technology Research Center) support program (NIPA-2009-(C1090-0902-0007)), the System Semiconductor Industry Development Center, Human Resource Development Project for IT SOC Architecture
Corresponding author: Hyeon-cheol YANG (blueecho@ssu.ac.kr)

2.3 Using object edges

Edge detection is widely used in object recognition. It uses filters, threshold values, or watershed algorithm. This method precisely recognizes real shape of objects, but it suffers from huge computation due to filtering or histogram operation.

2.4 Using difference image

This method[3-4] uses difference image between two images with some intervals but same background. It is simple in hardware implementation because it has only subtraction operation between previous image and current image.

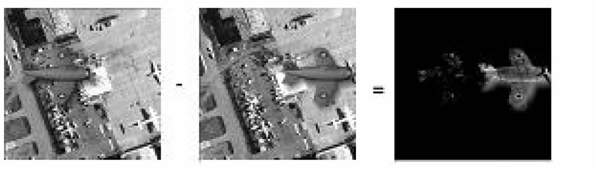

Fig. 1 Motion object detecting using difference of image

It is important to select an interval between two images because of velocity of objects. If the interval is short, the system cannot find slow objects, while it cannot find fast object if the interval is long. It is similar to full search method[5]. Block matching or fixed grid matching[6] is used for reducing computation, but accuracy quite decreases. Also noise appearing in difference image disturbs object detection.

3 Proposed architecture

A goal of proposed tracking unit is detecting a moving object with small size hardware and simple calculation. The proposed architecture based on difference image to reduce computation and support video encoding. It minimizes hardware area to be implemented on cheap FPGAs or SoCs instead of DSPs or embedded processors. It also achieves real-time operation because of reducing memory for current image because the proposed architecture detects not object shape but object movement.

3.1 Reduce memory

3.1.1 Use only luminance value

The input image data of motion tracker is 720×480 NTSC signals. Its signal consists of YCbCr or YUV signal. Proposed architecture recognize object using only Y value because Y(luminance) value is more sensitive than Cb, Cr or U,V(chrominance) value. Therefore, it is only Y value in memory. If input signal is 4:2:2 YCbCr format, it stores half of image data.

3.1.2 Store only previous image data

Although generating difference image process requires storage area for two image data, current and previous image, proposed architecture just useprevious data because real-time process requires no storage area for current data.

3.1.3 Sub-sampling

The proposed architecture exploits simple subsampling method to reduce spatial resolution. There are some subsampling methods, i. e. Average of several pixel and representative pixel. The proposed architecture exploits representative pixel with fixed sampling positions as shown in Fig. 2, since it requires no preprocessing for obtain sampling pixel. Proposed architecture exploits 4:1 subsampling to reduce memory size, because interlaced NTSC signal is easily half-sampled in y-axis.

10	20	30	40	50	60	70	80
11	22	33	44	55	66	77	88
13	24	35	36	46	47	58	58
15	26	23	76	37	48	45	45
22	33	44	45	76	65	56	55
23	34	45	56	67	78	89	99

10	30	50	70
13	35	46	58
22	44	76	56

Fig. 2 4:1 sub-sampling to reduce spatial resolution

There are some weak point, i. e. reduce accuracy, but an object that want to detect has some size, that is object is not a pixel. Proposed architecture uses these 3 methods for reduce memory. For example, if image resolution is 720×480 and 8 bits/pixel, it requires 2 073 kB for a frame in RGB format. It is just 87 kB for frame in proposed architecture.

3.2 Detecting object movement

3.2.1 Extract moving object area

In proposed architecture, an object is defined the area with separate from background and more than specified size.

A moving object generates difference image between N-th and ($N-m$)-th image in Eq.(1) and Fig.3.

$$\text{Difference}(I_n, I_{n-m}) = C\text{motion}_n. \quad (1)$$

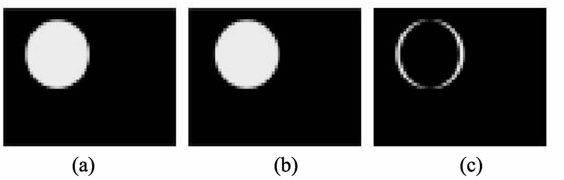

Fig. 3 Example of difference images: (a) I_n, (b) I_{n-m}, (c) C motion$_n$

There are some noise pixels in difference image, therefore the proposed architecture use threshold value. It uses profile accumulated from $C\text{motion}_n$ value of horizontal and vertical and selects moving object area that is larger than threshold value, as shown in Fig.4.

3.2.2 Using configuration factor

The proposed architecture adopts additional configuration factor to increase accuracy. These factors are sensitivity factor and minimum object size. Sensitivity factor value selects recognizing moving object how long it moves.

It helps to limit minimum speed of object. Minimum object size value helps recognizing moving object how large it is. N-skip value that is an interval between two

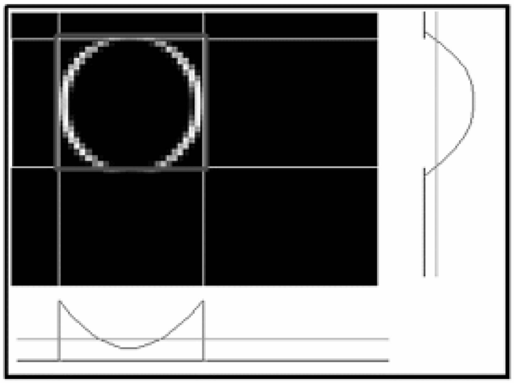

Fig. 4 Extract moving object area

images is related with velocity of moving object. Proposed architecture has two memories for previous image data. Each image data has different interval from current image. Fixed interval is weak to detect slow or fast object. Therefore proposed architecture has two variable interval modes, fast object mode and slow object mode.

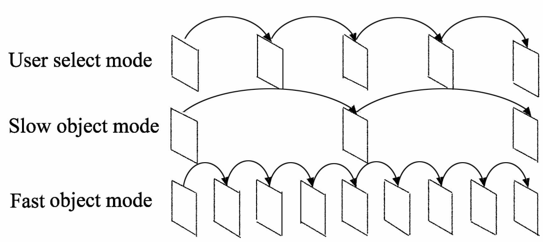

Fig. 5 N -skip mode for fast or slow object

Fig. 5 is comparison between each mode. The user select mode is skipping fixed number frame that is selected by user. If the movement of object is slow, it doesn't sense movement because movement value is smaller than sensitivity and user selected mode. In contrast, if the movement of object is fast, it doesn't catch object in current image. The proposed architecture calculates two movement values and selects larger value to detect slow and fast object. Moving object area detection algorithm in the proposed architecture consists of sampling, subtraction, profiling, comparison with threshold, comparing configuration factors and motion area extraction, as shown in Fig. 6.

The mov _ x and mov _ y is final value of motion tracking unit through calculate interval between center of moving bject area and center of frame, that is mov _ x and mov _ y is compensation value to track moving object.

This system actively tracks moving objects by doing effort to put moving object area into the center of frame. Motion tracking unit sends difference value between center of frame and center of moving object area. The value of mov _ x and mov _ y is difference output, as shown in Fig. 7. These values are sent to MCU and activate pan or tilt motor to compensate. Amount of each value indicates motor activation time. Motor activation according to the position of moving object area is shown in Fig. 8.

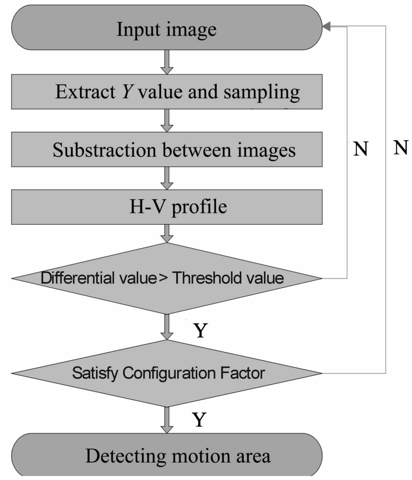

Fig. 6 Flow of moving area detection

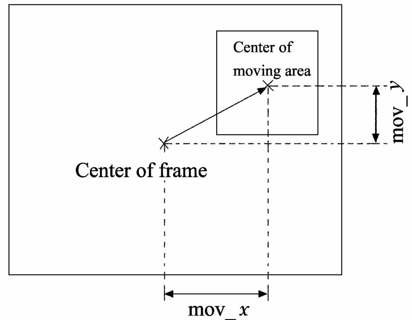

Fig. 7 Camera control parameters mov _ x and mov _ y

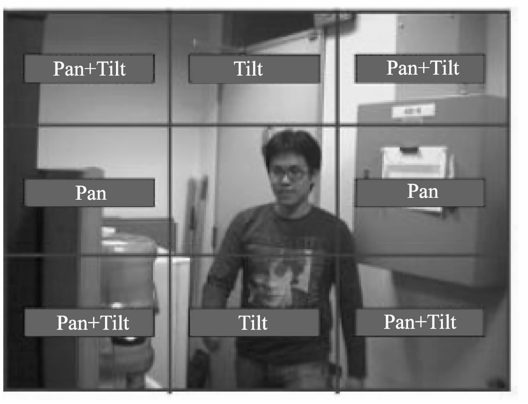

Fig. 8 Motor activation according to the position of moving object area

The operation of motion tracking unit is defined as Eq. (2)~Eq. (5).

$$ABS_{iuma} = | Cur_{iuma} - Ref_{iuma} |, \tag{2}$$

$$ABS_{iuma} \geqslant \text{threshold value}, \tag{3}$$

$$\text{Refresh} = \begin{cases} 1, & \text{when area} \geqslant \text{minimum size;} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$\text{mov}_{x,y} = \frac{\text{srt}_{x,y} - \text{end}_{x,y}}{2} - \text{center}_{x,y}, \tag{5}$$

Eq. (2) operates in process element. Eq. (3) and (4) operate in comparison unit. Eq. (5) operates in calculation unit.

3.2.3 Extendibility to motion estimation

In video compression, motion estimation examines the movement of objects in an image sequence to try to obtain vectors representing the estimated motion. Most of video compression standard including H. 264/AVC[7], MPEG-2, and MPEG-4 exploit motion estimation. Motion estimation is defined to find (m,n) that minimizes SAD (m,n) in Eq. (6). Eq. (6) consists of subtraction, absolute value, and accumulation those are same to motion tracking algorithm in Eq. (2) when $m = 1$. Therefore, motion estimator in video compression can share process elements with motion tracking unit. Since the proposed motion tracking unit performs some calculation for motion estimation in video compression, so it can be easily combined with surveillance system with video recording functionality based on video compression. Fig. 9 is a block diagram of H. 264/AVC motion estimator in Ref. [8], where the proposed motion tracking unit is embedded.

$$\mathrm{SAD}_{(m,n)} = \sum_{i=0}^{U-1}\sum_{j=0}^{V-1} \mid f_t(x,y) - f_{t-1}(x+\mathrm{d}y, y+\mathrm{d}y) \mid,$$
$$w \leqslant \mathrm{d}x, \mathrm{d}y \leqslant w,$$
$$x = U \times m + i, y = V \times n + j. \quad (6)$$

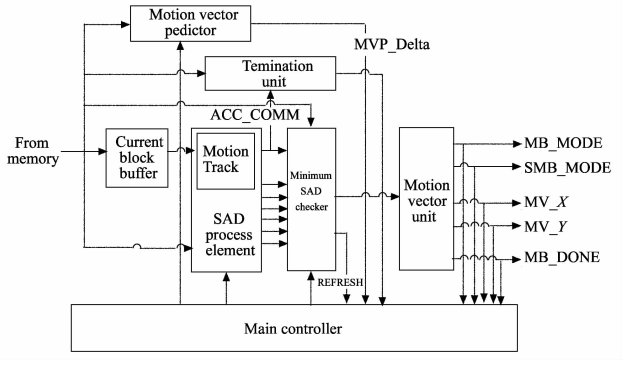

Fig. 9 H. 264/AVC motion estimators that embed the proposed motion tracking unit

Fig. 10 is a developed program with Microsoft visual C++ for proposed algorithm simulation.

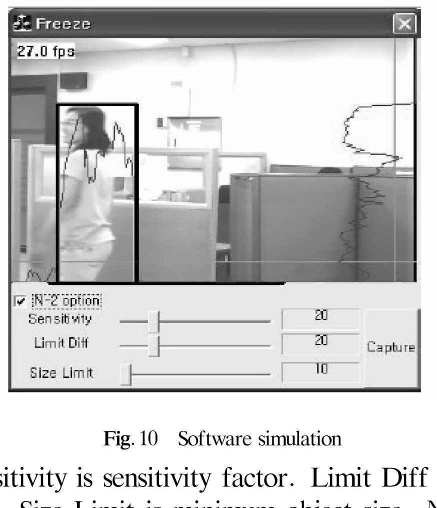

Fig. 10 Software simulation

Sensitivity is sensitivity factor. Limit Diff is threshold value. Size Limit is minimum object size. $N-2$ option is skipping 1 frame that is comparing n and $n-2$. It compare n and $n-1$ when that option unchecked. Input image data is NTSC signal (25 ~ 30 frames/sec, 720 × 480), but using data is only 360 × 240 image because of 4:1 sampling. Result of simulation is enough for detect movement of human and slow movement under 20 km/h.

4 Implementation

The proposed motion tracker unit consists of two frame memory, process element, comparing, position decision unit, calculation unit, I2C unit and controller. Fig. 11 is block diagram of motion tracker unit. Its gate count is about 25 000 gates, and its maximum operating frequency is 109.5 MHz. This unit is available to process VGA 48 frames/sec.

The proposed entire system is shown in Fig. 12. It consists of NTSC camera unit, pan-tilt motor, motor control unit, motion tracking unit, and PC for real-time surveillance and system configuration factor value with USB. The proposed motion tracker unit and USB interface unit is implemented in Vertex-4 xilinx FPGA (XC4VLX60).

The block diagram of tracking system is shown in Fig. 13. The input data of NTSC or PAL camera is 720 × 480 4:2:2 YCbCr or YUV signal. Motor unit has two motors for pan and tilt operation. It uses MCU (ATmega 16) for motor movement and I2C bus for communicating with other units because I2C bus is generally provided in many chip foundries. This system is enough for detect object under 20 km/h.

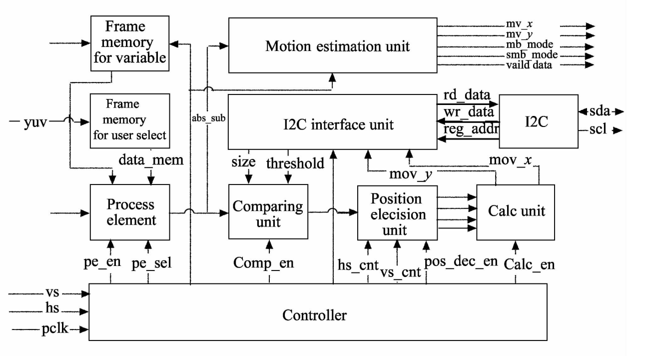

Fig. 11 Block diagram of motion tracker unit

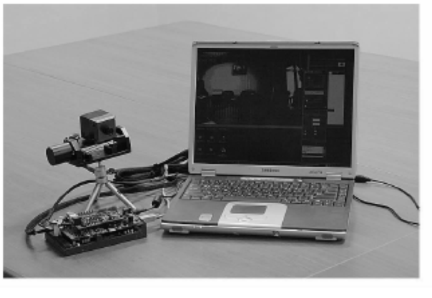

Fig. 12 Active observation camera systems

The core size without memory macro block is 830 × 305 μm with about 17 000 gates. The consuming power is about 61 mW at 110 MHz. This is very small size and consuming power than other complex method[9].

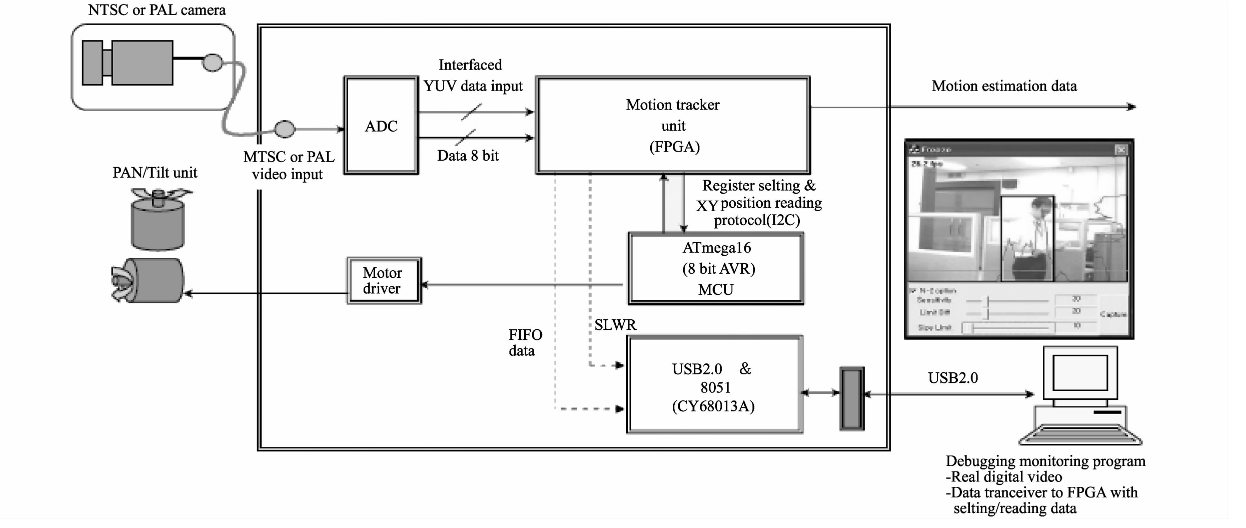

Fig. 13 Block diagram of active observation camera system

5 Conclusion

The proposed architecture based on differential images because of simple calculation and real-time operation. It uses three factors against noise and differential velocity. It reduces the required computation significantly, so it can be implemented on FPGAs or SoCs instead of expensive DSPs or embedded processors. Surveillance systems with cameras, in general, almost store image or video data. This architecture has two ways to reduce storage data. One is just storing still image of moving object area because background of images are same in camera's angle. Other is storing video data with video encoding method, like H.264 or MPEG, because proposed architecture includes motion estimator for video encoding, so it can beeasily combined with surveillance system with video recording functionality based on video compression to store data for long times. This paper proposes cost-effective architecture for simple, fast and small. It helps to integrate other unit for SoCs.

References

[1] S. Bhat, M. Saptharishi, P. K. Khosla, 2000. Motion dectection and segmentation using image mosaics. *IEEE International Conference on Multimedia and Expo*, 3: 1577-1580.

[2] Comaniciu, V. Ramesh, 2000. Mean shift and optimal prediction for effcient object tracking. *IEEE International Conference on Image Processing*, 3: 70-73.

[3] Murray, A. Basu, 1994. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5): 449-459.

[4] K. W. Lee, S. W. Ryu, S. J. Lee., K. T. Park, 1998. Motion based object tracking active camera. *Electronic Letters*, 34(3): 256-258.

[5] Seung Hyun Nam, Jong Seob Beak, Tae Young Lee, Moon Key Lee, 1994. A VLSI design for full search block matching motion estimation. IEEE Annual ASIC Conference and Exhibit, p. 354-257.

[6] L. Marcenaro, G. Vernazza, C. S. Regazzoni, 2001. Image stabilization algorithms for video-surveillance applications. IEEE International Conference on Image Processing, p. 349-352.

[7] ITU-T Rec, 2005. H.264, Advanced Video Coding for Generic Audio Visual Services.

[8] Yeon-cheol Yang, Seong-soo Lee, 2005. Low-power low-computation motion estimator architecture for H.264. Information Technology System-on-Chip Conference, p. 686-689.

[9] W. Adawy, M. Bayoumi, 2007. Low power VLSI prototype for motion tracking architecture. ASIC/SOC Conference IEEE Press: p. 57-64.