Designing Video Processing Surveillance System

EE382 Embedded Software Systems

Literature Survey

Koichi Sato

Abstract

The objective of this project is to design the embedded intelligent surveillance system. That is, to turn our PC based research into embedded system. Our previous system tracks persons and recognizes two-person interactions using grayscale monochrome side-view image sequence captured by a single static camera. Different from multimedia system, the output of surveillance system is not always the image sequence, but also the human positions, image content features and so on. Therefore, the architecture of video surveillance system differs greatly depending on their objective and outputs. In this survey, I learned several embedded surveillance system architectures proposed by several forerunners. As reference of these architectures, I propose a rough architecture of embedded surveillance system applicable to our algorithms.

Introduction

Recognizing objects using video sequence is an important topic in computer vision field and has a variety of potential applications. Many researchers have proposed actual applications with this problem. In [1], A. Pentland proposed “wearable device” that sees the people using image sensor and understand the environments so that computer can act or respond appropriately without detailed instructions. In [2], Mahonen proposed wireless video surveillance system concepts using intelligent surveillance camera that consists of digital camera, CPU or DSP for image processing and wireless radio modem. Several other researchers [3]-[6] also proposed varieties of embedded image processing systems. Even in systems combined with PC, many of the applications embed the image processing part in their camera modules. Some of the reasons are: i) transforming a huge amount of video data is hard to achieve or expensive; ii) size or weight restriction of the component; and iii) real time solution is needed.

In [10], Shirai at el. proposed real time surveillance system using TI DSP chips TMS320C40. The system consists of several boards connecting in series each other and is

capable of computing optical flow calculation with floating point output more than 30 frames per second. In this system the DSP chip located between two memories computes and transfers the image data from the video memory to the other memory that is connected to the next process stage.

In my previous research [7],[8], we have proposed a real-time surveillance system that recognizes human interactive activities using side-view image sequence. The system recognizes humans based on the pixel velocities extracted using "temporal spatio-velocity transform". The system consists of a single monochrome video camera, a video capturing device and a PC. When it turns to an actual application, the system would use several cameras due to the small observation range of the single system, and also several problems happen: (i) several PC is required because only one or two camera per PC can be manageable; and (ii) synchronization between small system sets are hard design.

To overcome the problem (i), I propose a video-processing camera system that the fundamental image processing part (that is human segmentation processing) is embedded in the camera part. In our system, there are both processes good for embedded system and PC-based system. Therefore, the proposed system consists of video camera and a processor such as DSP or CPU, segments humans and transmits human positions, velocities and sizes to PC (transmitting data is small). (PC will process later complex part.)

	Embedded image processing	PC based image processing
Merit	- Cost effective - Effective for pipeline-type process	- Good for complex process
Demerit	- Expensive for a process using large amount of data (like database search)	- Camera-PC transmission data is huge -> The number of connectable cameras to a PC is restricted

Table 1. Embedded image processing vs. PC based image processing

Design Target

The design target is an embedded part of the hybrid intelligent surveillance system that some of the image processing is done by the embedded system and the other image process is done by PC system. This is one of the solutions for the problems when our PC based research is applied to practical applications, which are excessive amount of transferring data, high computation requirement and more than one area sensor requirement. This system achieves good cost performance as well as good computation

performance as a whole system by separating processes into processes should be done by embedded system and ones should be done by PC system.

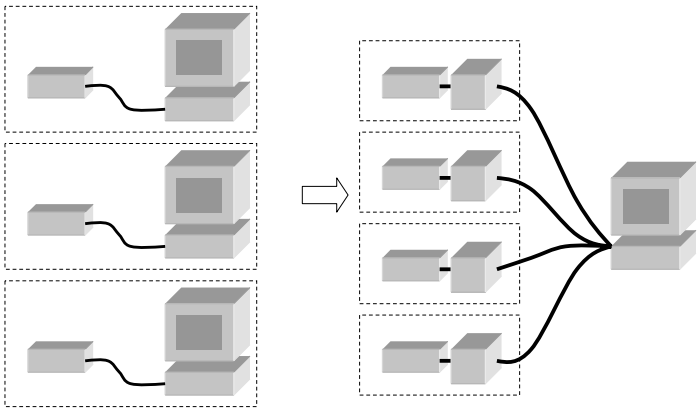

Figure1. The “one camera per PC” system vs. “many cameras per PC” system.

Algorithm Overview

To construct the architecture, the processes should be considered.
The system we proposed before detects and recognizes two-person interactions. Using grayscale single side view camera fixed beside the sidewalk, it tracks the humans moving the sidewalk in front of the camera. By analyzing the trajectory patterns of two persons, it outputs the interaction moment and its type.
Figure2 is the overview of our system. Here, I discuss mainly on the computation in each stage.

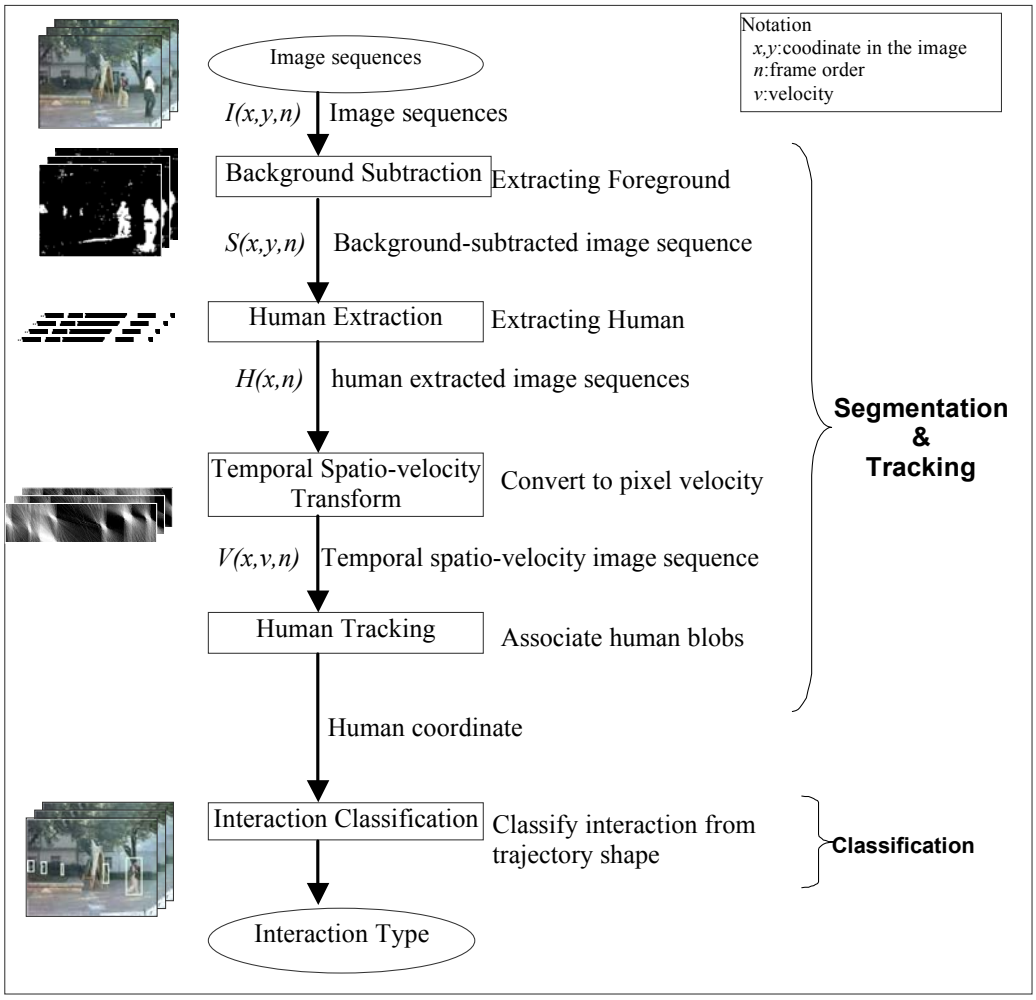

Figure2. System overview

Background Subtraction

We use simple background subtraction and binarization to segment foreground region using threshold *Th*,

$$S(x,y) = \begin{cases} 1 & |I(x,y) - B(x,y)| > Th \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where *x*,*y* is the horizontal and vertical coordinate of the images, *S*(*x*,*y*) is the binarized image, *I*(*x*,*y*) is the original image, *B*(*x*,*y*) is the stored background image.

Threshold *Th* is computed each frame so as to be the appropriate value. The threshold in the *n*th frame *Th*(*n*) can be obtained using the threshold in the previous frame *Th*(*n*-1) and the number of isolated pixels in the *n*th frame N_i.

$$Th(n) = Th(n\text{-}1) + \tau \times (N_i\text{-}N_t) \quad (2)$$

where N_t is the target number and τ (>0) is the learning coefficient. Through iterative computations, the threshold Th converges to a value that generates a binary image $S(x,y)$ whose number of isolated pixels N_i is same as the target number N_t.

Object Extraction

Object extracting process is performed by vertical projection of blobs within Region A and re-binarize the projection value by threshold T_H.

$$H(x) = \begin{cases} 1 & \sum_{y=a}^{b} S(x,y) > T_H \\ 0 & otherwise \end{cases} \quad (3)$$

where T_H is a threshold, which is constant in all situations and $H(x)$ is the object extracted binary image. Region A is set slightly above and below the horizon line that is expressed as:

$$\text{Region A: } a < y < b \quad (4)$$

Temporal Spatio-velocity transform (TSV transform)

We proposed TSV transform in the previous paper [7],[8]. TSV transform extracts the pixel velocity from binary image sequences. Here, we use one-dimensional binary image sequence coming from object extraction.

$$V_n(x,v) = e^{-\lambda} V_{n-1}(x-v,v) + (1-e^{-\lambda}) H_n(x) \quad (5)$$

To group the pixels with similar velocity, we binarized the temporal spatio-velocity image by a fixed threshold Th_v.

$$\widetilde{V}_n(x,v) = \begin{cases} 1 & \text{if } V_n(x,v) \geq Th_V \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Human blobs are obtained by performing the connected component region labeling operation.

Human Tracking

Five features of each blob are used for associate blobs over frames. These are the horizontal size of the blob (S_h), blob area size (S_a), vertical texture (χ) and the horizontal projection of the binary blob (p). The difference values S_a',S_h',χ',p' from the tracking features are then calculated, as well as the acceleration from the tracking blob (a). These variables are independent and are measures of similarity to the tracking blob. These variables are computed as follows:

$$
\begin{aligned}
S_h^{'} &= S_h - S_{h\text{-}tracking} \\
S_a^{'} &= S_a - S_{a\text{-}tracking} \\
\chi' &= \sum_y \left| \chi \cdot p - \chi_{tracking} \cdot p_{tracking} \right| \\
p' &= \sum_y \left| p - p_{tracking} \right| \\
a &= 2\frac{x - x_{tracking}}{(n - n_l)^2} - 2\frac{v_{tracking}}{n - n_l}
\end{aligned}
\tag{30}
$$

where $S_{htracking}$ is the horizontal size of the tracking human blob, $S_{atracking}$ is the area size of the tracking human blob, $\chi_{tracking}$ is the horizontally projected texture of the tracking blob, $\boldsymbol{p}_{tracking}$ is the horizontally projected binary blob of the tracking blob, n_l is the frame number when a last blob is selected to the tracking blob, $v_{tracking}$ is the velocity of tracking blob of n_lth frame, $x_{tracking}$ is the horizontal position of the tracking blob of n_lth frame.

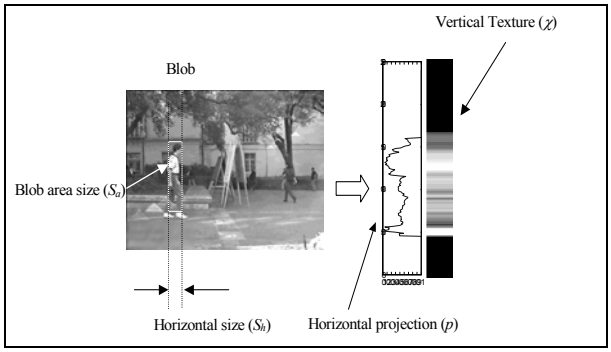

Figure3. The features for identification

Interaction Classification

Two-person interaction is recognized using the trajectory of each person. The features are evaluated with Nearest Mean Method class features in the database. This computation is complex and needs database access, so this part should be PC based system rather than embedded system.

Embedded System Designing

Assignment some processes to embedded system

I would like to assign the processes of the current system to embedded system and PC from two aspects, data-transferring amount and calculation amount. So, first, let me explain about the current algorithm overview.

As seen in Figure4, I assigned notation A-H to each process and also assigned notation 1-9 to each data transfer rate. Now I consider the best architecture for embedded system based on the different aspects.

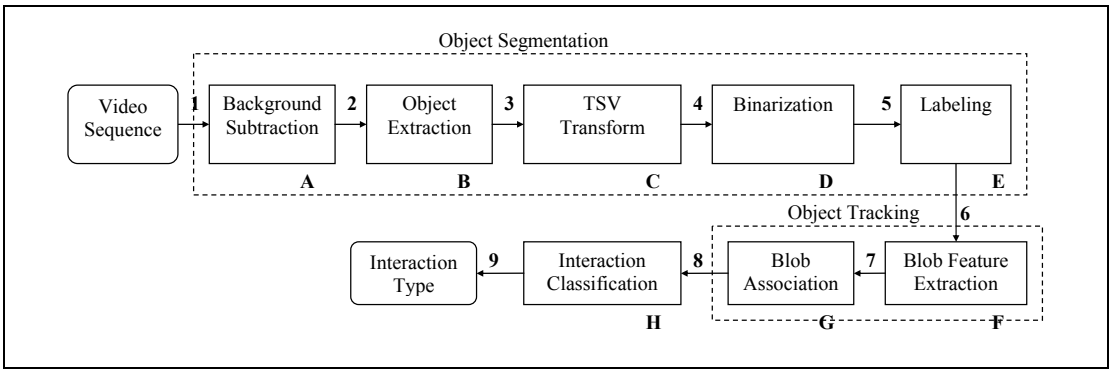

Figure4. System Structure

Data Transfer Rate Consideration

Let us consider the data transfer amount between steps. In this consideration, I assumed NTSC monochrome format video signal are captured as sequence of VGA (640x480) size image. Table 2 is the specification of the NTSC format.

Frame rate	30Hz (Monochrome)/29.97Hz (Color)
Number of raster	525 lines
H-Sync Frequency	15750Hz
Aspect ratio	0.75

Table 2. NTSC format

Based on the format, I calculate the data transfer rate for the current system.

number	Data Type	Transfer rate
1	8bit Grayscale image (658x525x30 Hz)	98.72 MBPS
2	1bit Binary image (658x525x30 Hz)	12.34MBPS
3	1bit Binary one line image (658x1x30 Hz)	18.75kBBS
4	8bit Grayscale Image (658x40x30 Hz)	7.52MBPS

5	1bit Binary Image (658x40x30 Hz)	940kBPS
6	8bit labeling map (658x40x30 Hz)	7.52 MBPS
7	Blob features (16bit x8 x20x30 Hz)	76.8kBPS
8	Blob trajectories (16bit x2x100x20x30 Hz)	1.23MBPS

Table 3. Data transfer rate

Computation Consideration

Also, I consider the computation amount in each step. The computation time represents addition/subtraction/multiplication times per seconds.

Step	Description	Computation time
A	Subtraction and comparison for entire image	18.4M [times/sec]
B	Addition for specific region and comparison one line	780k [times/sec]
C	Parallelogram shift and addition for entire TSV image	2.3M [times/sec]
D	Comparison for entire TSV image	768k [times/sec]
E	Labeling process for Binary TSV image	≈ 1.5M [times/sec]
F	Calculation of blob features	≈ 380k [times/sec]
G	Associating blob using features	≈ 57k [times/sec]
H	Activity recognition	-

Table 4. Estimated Computation

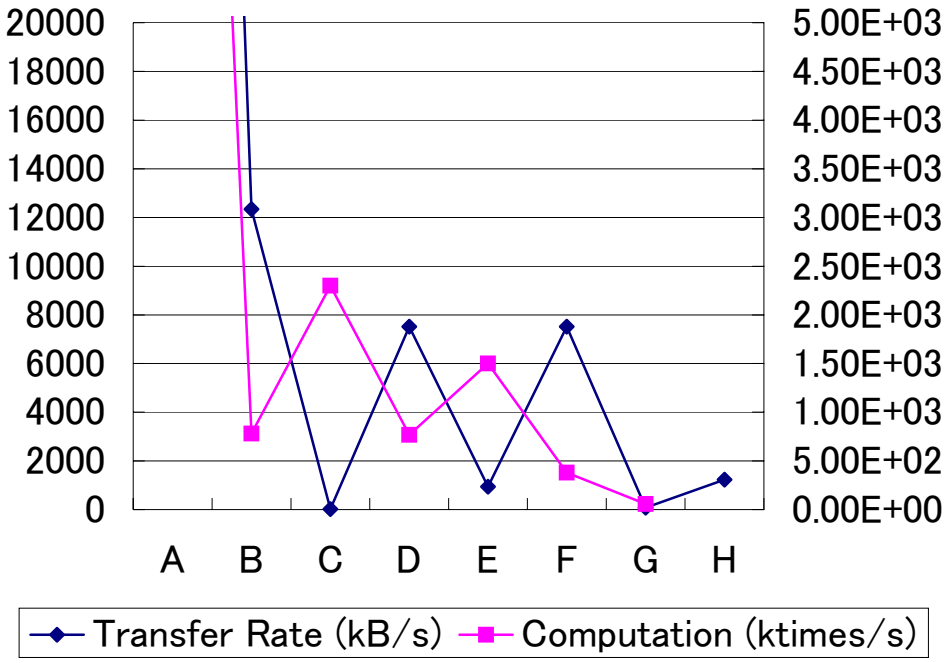

Figure5. The transfer rate and computation amount of each step

Congeniality to embedded system

Among the processes above, some of them are congenial to the embedded system and some are not. Generally, pipeline-like process, simple but large computation process is good for embedded system. I ranked each process regarding the congeniality to embedded system.

Step	Description	Congeniality
A	Subtraction and comparison for entire image	◎
B	Addition for specific region and comparison one line	◎
C	Parallelogram shift and addition for entire TSV image	◎
D	Comparison for entire TSV image	◎
E	Labeling process for Binary TSV image	○
F	Calculation of blob features	○
G	Associating blob using features	○
H	Activity recognition	△

Table 5. Congeniality of each process to embedded system

Process Assignment

Regarding the computation amount, A-E is large computation processes and F-H is comparatively computation processes. Regarding data transfer rate, 3 and 5 are the small data transfer rate. Therefore, the overall system is efficient when step A to D are assigned to the embedded system and step E to H are assigned to the PC. Considering several aspects, I conclude to assign A-D to the embedded system. (Figure6)

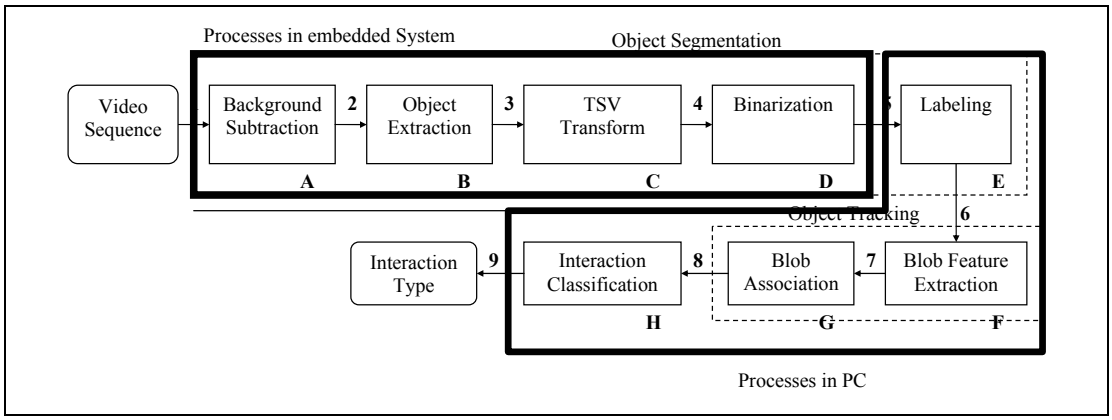

Figure6. Process Assignment

Hardware structure

Figure7 is the rough structure of this embedded system. The video signal from the camera is AD converted with a pixel clock generated clock generator. The digitized image data are first captured in the DRAM memory specialized for image and DSP processor carries the image data to the second image DRAM memory or FIFO memory with several basic computations. The computed data is obtained by CPU and is transferred to PC with small computations.

The video signal from camera also goes to sync-signal separator unit and /HD (horizontal sync-signal and /VD (vertical sync-signal) is separated. /HD signal are used for generating the clock in PLL (Phased Locked Loop) unit by comparing the phase between /HD and pseudo-/HD, that is down-clocked from the generated clock.

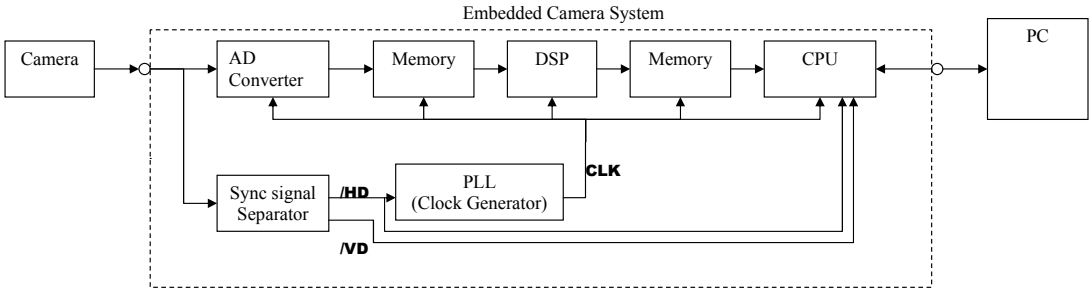

Figure7. Rough hardware structure

References

[1] Pentland, A. “Looking at people: sensing for ubiquitous and wearable computing” Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 22 Issue: 1, Jan. 2000, Page(s): 107 -119

[2] Mahonen, P., “Wireless video surveillance: system concepts”, Image Analysis and Processing, 1999. Proceedings. International Conference on , 1999 , Page(s): 1090 -1095

[3] Hougen, D.F.; Benjaafar, S.; Bonney, J.C.; Budenske, J.R.; Dvorak, M.; Gini, M.; French, H.; Krantz, D.G.; Li, P.Y.; Malver, F.; Nelson, B.; Papanikolopoulos, N.; Rybski, P.E.; Stoeter, S.A.; Voyles, R.; Yesin, K.B., “A miniature robotic system for

reconnaissance and surveillance**",** Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on , Volume: 1 , 2000, Page(s): 501 -507 vol.1

[4] Adler, E.; Clark, J.; Conn, M.; Phuong Phu; Scheiner, B. "Low-cost technology for multimode radar", IEEE Aerospace and Electronics Systems Magazine, Volume: 14 Issue: 6 , June 1999
Page(s): 23 -27

[5] Marcenaro, L.; Oberti, F.; Foresti, G.L.; Regazzoni, C.S., "Distributed architectures and logical-task decomposition in multimedia surveillance systems", Proceedings of the IEEE , Volume: 89 Issue: 10 , Oct. 2001, Page(s): 1419 -1440

[6] Soatto, S.; Frezza, R.; Perona, P., "Motion estimation via dynamic vision**"** Automatic Control, IEEE Transactions on , Volume: 41 Issue: 3 , March 1996 Page(s): 393 -413

[7] K.Sato and J.K.Aggarwal, "Tracking and Recognizing Two-person Interaction in Outdoor Image Sequences", 2001 IEEE Workshop on Multi-Object Tracking, pp.87-pp.94, Vancouver, CA, July, 2001.

[8] K.Sato and J. K. Aggarwal, "Tracking objects using temporal spatio-velocity transform", *2001 IEEE Workshop on PETS*, Kauai, Hawaii, December, 2001.

[9] Y. Nara and S. Nagasaka, "Basic transistor TV textbook", *Ohm Publication*, Japan.

[10] Shirai, Y.; Miura, J.; Mae, Y.; Shiohara, M.; Egawa, H.; Sasaki, S, "Moving object perception and tracking by use of DSP", *Computer Architectures for Machine Perception*, 1993. Proceedings , Nov. 1993, Page(s): 251 -256