Developing a Real-Time Person Tracking System Using the TMS320C40 DSP

APPLICATION BRIEF: SPRA189

Authors: Ryuzo Okada
Shinya Yamamoto
Yasushi Mae
Advising Professor: Yoshiaki Shirai
Department of Mechanical Engineering for Computer Controlled Machinery, Osaka University

Digital Signal Processing Solutions
January 1996

Texas Instruments

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

Figures

Developing a Real-Time Person Tracking System Using the TMS320C40 DSP

Abstract

This application report describes the development of a real time system for tracking a walking person using a high-speed image processor and an active camera head. The image processor is composed of five digital signal processing (DSP) boards. Each DSP board includes two Texas Instruments (TI™) TMS320C40 DSPs that communicate with each other through the communication ports and perform various image-processing in parallel.

The first three DSP boards calculate flows (the velocity field of an image) based on the gradient method. Next, the region of a person on the image is extracted from the optical flows using knowledge about the shape of a person. Finally, the camera head is controlled to keep the person in the field of view of the camera. The result is shown for tracking a person in a cluttered background.

This document was an entry in the 1995 DSP Solutions Challenge, an annual contest organized by TI to encourage students from around the world to find innovative ways to use DSPs. For more information on the TI DSP Solutions Challenge, see TI's World Wide Web site at www.ti.com.

World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

Introduction

Collision avoidance is a key consideration for people working with autonomous robots. Computer vision is an effective tool in collision avoidance because it tracks a person without requiring physical contact.

Tracking generally takes excessive time to process images because of the enormous amounts of data contained in the images. A camera must be able to turn to keep a person in its field of view while the person moves in the environment. Our objective is to track a walking person in real time using an active vision system with a camera head and high-speed image processor.[1] We use *optical flows*, the velocity field of an image, to detect a person in an image.

Various methods are available to calculate the optical flows.[2] [3] This application report uses the gradient method.[4] [5] Two different spatial filters are convolved to an input image to solve the constraint equation of the gradient method. Though the cost of the convolution operation can be prohibitive, dividing the two-dimensional spatial filters into two one-dimensional recursive filters can reduce the processing.[6] The region of the person is segmented based on the difference in velocity between the person and the background. However this method cannot detect the correct region of the person if the velocity of the person is similar to the background velocity or a part of the person’s body is occluded.

We use the knowledge of the shape of the person to cope with these situations. The shape of the person is the ratio of the height to the width of the person in the image. Since the ratio of the walking person does not change much, the region of the person is determined to keep the ratio by modifying the region detected from the optical flows. The active camera head is controlled to keep the person in the field of view of the camera.

The image processor has a video board and five DSP boards. Each DSP board contains two TI TMS320C40 DSPs that communicate with each other through the communication ports. The video board is the interface between the active camera and the host computer. The DSP boards perform image processing. Our tracking is implemented to the image processor. The first three DSP boards calculate the optical flows. The fourth DSP-board detects the person and controls the active camera head . And the fifth DSP-board makes an image of the result for display.

Section 1 describes the extraction of optical flows. The result of extraction of optical flows is shown.

Section 2 shows the tracking algorithm.

Section 3 describes the implementation of the tracking algorithm to the image processor. The architecture of the image processor is shown. The result of tracking is shown.

Extraction of the Optical Flows

The velocity field is derived from optical flows calculated using the gradient method. The gradient method is based on spatio-temporal filtering.[1 2 3]

Constraint Equation of Gradient Method

Let $f(x, y, t)$ denote a brightness at a point (x, y) in an image at time t. If this point moves to a point $(x + dx, y + dy)$ at time $t + dt$, the following equation holds:

$$f(x,y,t) = f(x+dx,y+dy,t+dt) \quad (1)$$

Taylor expansion of the right side of the equation (1) is

$$f(x,y,t) = f(x,y,t) + f_x(x,y,t)dx+f_y(x,y,t)dy + ft(x,y,t)dt + e, \quad (2)$$

where

$f_x = \frac{\partial f}{\partial x}, f_y = \frac{\partial f}{\partial y}, f_t = \frac{\partial f}{\partial t}$, and e is high order terms of dx,dy,dt.

Assuming that e is negligible, we obtain the next equation:

$$f_{x\,=}\,(x,y,t)u + f_y\,(x,y,t)v + f_t(x,y,i) = 0 \quad (3)$$

where $u = \frac{dx}{dt}, v = \frac{dy}{dt}$ and u and v are the x and y component of the velocity.

This is the constraint equation of the gradient method. However, equation (3) has two unknowns (u and v) and cannot be solved. The unknowns u and v are calculated as shown in the *Filtering* section.

Filtering

Two different spatial filters g,h are convolved to the input image $f(x,y,t)$, and the temporal smoothing filter p is applied using these two filtered images. The following two constraint equations are derived according to equation (3).

$$(p * g * f)_x u + (p * g * f)_y v + (p * g * f)_t = 0 \quad (4)$$

$$(p * h * f)_x u + (p * h * f)_y v + (p * h * f)_t = 0 \quad (5)$$

where

* represents convolution.

u and v are calculated from equations (4) and (5) as follows:

$$u = \frac{(p*g*f)_t(p*h*f)_y - (p*h*f)_t(p*g*f)_y}{(p*g*f)_x(p*h*f)_y - (p*g*f)_y - (p*h*f)_x} \quad (6)$$

$$v = \frac{(p*h*f)_t(p*g*f)_x - (p*g*f)_t(p*h*f)_x}{(p*g*f)_x(p*h*f)_y - (p*g*f)_y(p*h*f)_x} \quad (7)$$

If the absolute value of denominator of equations (6) and (7) is small, the velocity at such point is not calculated. This is because the independence of equations (4) and (5) is insufficient.

These filters must have the following three characteristics.

- To assure differentiability
- To suppress the high-order terms of *dx*,*dy*,*dt*
- To make two independent constraint equations (4) and (5)

We use the horizontally differentiated gaussian filter for *g*. The horizontal variance of this gaussian is larger than the vertical variance, as shown in equation (8). Similarly, for the filter *h*, the vertically differentiated gaussian filter. The vertical variance is larger than the horizontal variance, as shown in equation (9).

$$\begin{aligned} g(x,y) &= \frac{\partial}{\partial x}\left[\frac{1}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{1}{2}\left\{\left(\frac{x}{\sigma_1}\right)^2 + \left(\frac{y}{\sigma_2}\right)^2\right\}\right)\right] \\ &= \frac{x}{\sqrt{2\pi}\sigma_1^3}\exp\left(-\frac{x^2}{2\sigma_1^2}\right)\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{y^2}{2\sigma_2^2}\right) \end{aligned} \quad (8)$$

$$\begin{aligned} h(x,y) &= \frac{\partial}{\partial y}\left[\frac{1}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{1}{2}\left\{\left(\frac{x}{\sigma_2}\right)^2 + \left(\frac{y}{\sigma_1}\right)^2\right\}\right)\right] \\ &= \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{y^2}{2\sigma_2^2}\right)\frac{y}{\sqrt{2\pi}\sigma_1^3}\exp\left(-\frac{x^2}{2\sigma_1^2}\right) \end{aligned} \quad (9)$$

where

σ_1, σ_2 are variances and $\sigma_1 > \sigma_2$.

Equation (10) is used for the temporal filter *p*

$$y(i) = \beta_0 x(t) + \beta_1 x(t-1) + \beta_2 x(t-2) + \ldots + \beta_n x(t-n) + \ldots \quad (10)$$

where

$y(t)$ is the output.

$x(t)$ is the input image at time t. The coefficients hold the following conditions:

$$\sum_{i=0}^{n} \beta_i \quad , \quad \lim_{n \to \infty} \beta_n = 0$$

Recursive Filtering

Spatial Filter

The convolution operation of a two-dimensional spatial filter is computationally expensive unless the filter is small. A spatial filter with size M x N is convolved, the number of calculations is M x N at a pixel. Equations (8) and (9) show it is possible that the filter g,h can be separated to x-directional one-dimensional filter and y-directional one-dimensional filter, and they can be convolved separately. This can reduce the calculation to $M + N$ at a pixel. However, this convolution operation is still computationally expensive. This section describes how the recursive filtering method can reduce the calculation more.[6]

Let $y(i)$ denote the output of a filter $h(i)$ and $x(i)$ denote the input to the filter.

$$y(i) = \sum_{k=0}^{N-1} h(k)x(i-k) \tag{11}$$

Equation (12) is the transfer function, or Z transformation, of equation (11).

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k} \tag{12}$$

The next equation is an approximation of this transfer function (12).

$$H_a(z) = \frac{\sum_{k=0}^{m-1} b_k z^{-(k-1)}}{1 + \sum_{k=1}^{n} a_k z^{-k}} \tag{13}$$

The next equation is the inverse Z transformation of (13). This equation shows that filtering can be performed recursively.

$$y(i) = \sum_{k=0}^{m-1} b_k x(i-k) - \sum_{k=1}^{n} a_k y(i-k) \tag{14}$$

The combination of equations (16) and (18) construct the two-dimensional filter *g* and *h*. In the case of filter *g*, *x*-directional filter is differential gaussian filter (18) and *y*-directional filter is gaussian filter (16).

Gaussian Filter

The next equation is an approximation of the Gaussian filter.

$$S(n) = k(\alpha |n| + 1)e^{-\alpha|n|} \tag{15}$$

where

$$k = \frac{(1-e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}$$

and α decides the variance σ^2 of Gaussian. This approximate filter can be represented in a recursive form based on equations (11) through (14) as follows:

$$y(n) = y_1(n) + y_2(n) \quad n = 1,.., M \tag{16}$$

$$y_1(n) = k[x(n) + e^{-\alpha}(\alpha - 1)x(n-1)] + 2e^{-\alpha}y_1(n-1) - e^{-2\alpha}y_1(n-2) \quad n=1,..,M$$

$$y_2(n) = k[e^{-\alpha}(\alpha - 1)x(n+1) - e^{-2\alpha}x(n+2)] + 2e^{-\alpha})\, y_1(n+1) - e^{-2\alpha}y_1(n+2) \quad n = M,..,1$$

Differential Gaussian Filter

Equation (17) is an approximation of the differential Gaussian filter.

$$D(n) = kne^{--\alpha|n|} \tag{17}$$

where,

$$k = \frac{(1-e^{-\alpha})^2}{e^{-\alpha}}$$

α is constant, which decides filter size. Equation (17) is represented in recursive form based on equations (11) through (14) as follows:

$$y(n) = ke^{-\alpha}[y_1(n) - y_2(n)] \quad n = 1, \cdots, M \tag{18}$$

$$\begin{aligned} y+(n) &= ke^{-\alpha}[x(n-1) + 2e^{-\alpha} y + n(n-1) - e^{-2\alpha} y + (n-2)] \\ &= ke^{-\alpha} y1(n) \quad n = 1, \cdots, M \\ y-(n) &= -ke^{-\alpha}[x(n+1) + 2e^{-\alpha} y - (n+1) - e^{-2\alpha} y - (n+2)] \\ &= -ke^{-\alpha} y_2(n) \quad n = M, \cdots, 1 \end{aligned}$$

Temporal Filter

The temporal filter shown in equation (10) is also represented in recursive form. Equation (19) is a recursive temporal filter.

$$y(t) = \beta x(t) + (1 - \beta) y(t - 1) \tag{19}$$

where

$x(t)$ is an input image.

$y(i)$ is the output of the filter.

β determines the filter size.

Result of Extracting Optical Flows

Figure 2 shows the extracted optical flows from the input images shown in Figure 1. The constant α in equations (16) and (18) and β in equation (19) are fixed, as shown in this experiment:

$$\alpha_x^g = 0.3, \quad \alpha_y^g = 0.7$$
$$\alpha_x^h = 0.7, \quad \alpha_y^h 0.3$$
$$\beta = 0.5$$

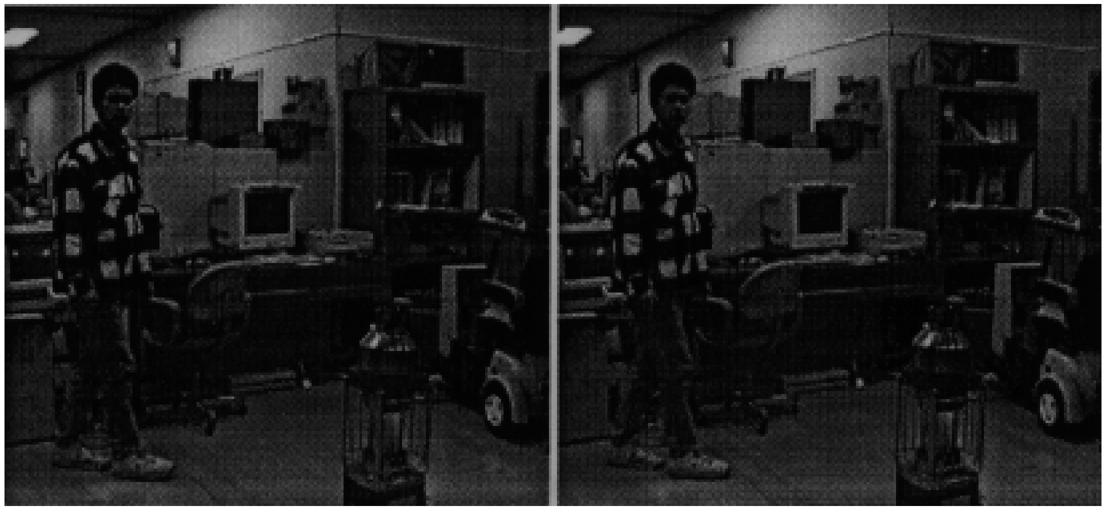
where

subscripts *x* and *y* mean the directions of the filters.

superscripts *g* and *h* mean the spatial filters *g* and *h*, respectively.

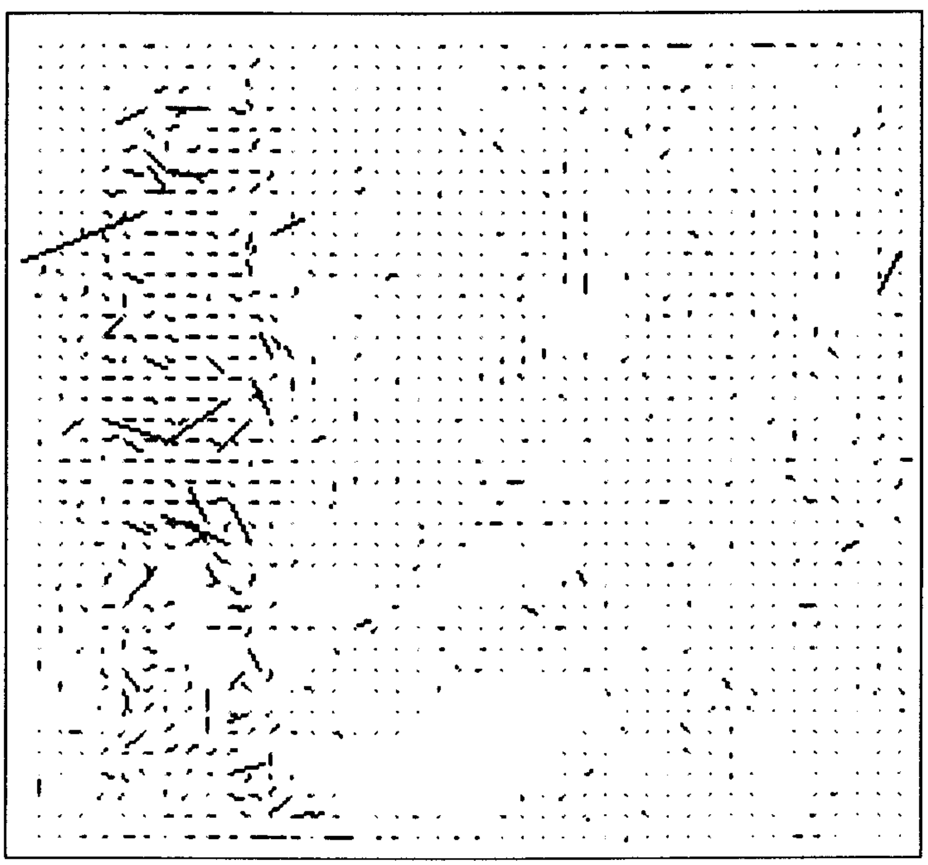
If the absolute value of the denominator in equations (6) and (7) is less than a threshold, the optical flow is not extracted in such pixels because the extracted optical flow is not reliable. In this experiment, the threshold is 1.0.

Figure 1. Input Images



time *t* time *t+ 1*

Figure 2. Optical Flows



Person Tracking Algorithm

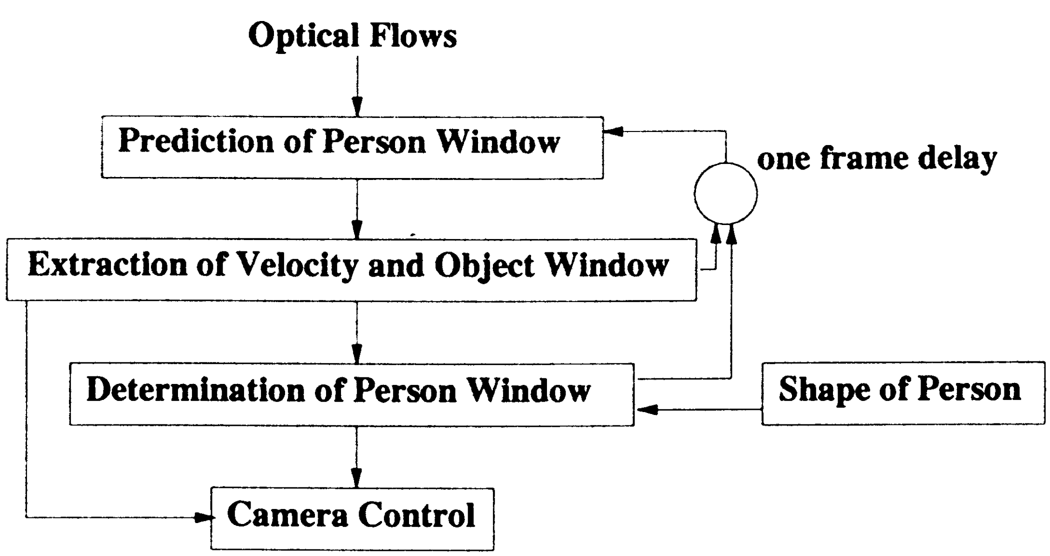
Detection of a person is accomplished by circumscribing the person with a rectangle, or *person window*, in the image. The tracking determines the size and location of the person window over a sequence of images. It is important to control the active camera head to keep the person in the field of view of the camera.

Person detection is performed based on the optical flows. The region of the person is segmented based on the differences of the velocities between the person and the background.

When the velocity of the person is similar to the background velocity, it is difficult to segment the region of the person. When part of the body is occluded, the whole body can not be segmented. In this case, the knowledge of the shape of the person is used to enable to track the person. When a person is walking, his shape in the image does not change greatly; that is, the ratio of the height to the width of the person window does not change much. The person window is determined to keep this ratio. This method can overcome the difficulties stated below.

Figure 3 shows the tracking algorithm.

Figure 3. Tracking Algorithm



Person-Window Prediction

Let (u_{t-1}, v_{t-1}) denote the velocity of the person, and W_{t-1} denote the person window at previous frame (time $t - 1$). W_{t-1} has two parameters: location and size. The location is represented by the center point of the window (x_{t-1}, y_{t-1}), and the size is represented by the height and the width of the person window ($height_{t-1}$, $width_{t-1}$).

The interval between frames is short enough that we can assume the person maintains velocity (u_{t-1}, v_{t-1}) as he moves. The predicted person window $W_t^{predict}$ at current frame (time t) is obtained by shifting W_{t-1} by the velocity (u_{t-1}, v_{t-1}). The location of $W_t^{predict}$ is ($x_{t-1} + u_{t-1}$, $y_{t-1} + v_{t-1}$) and the size is ($height_{t-1}$, $width_{t-1}$).

Extraction of Velocity and Object Window

The mean velocity (u_{t-1}, v_{t-1}) is calculated in the predicted person window $W_t^{predict}$ and is the velocity of the person at time t. The region that has the velocity similar to the mean velocity (u_{t-1}, v_{t-1}) is extracted near the predicted person window $W_t^{predict}$. Velocities u and v, satisfying the following conditions (20) and (21), are the similar velocities to the mean velocity (u_t, v_t). The window circumscribing this region is called *object window*.

$$u_{min} < u < u_{max} \qquad (20)$$

$$v_{min} < v < v_{max} \qquad (21)$$

where

$$u_{min} - u_t + c_1|u_t| - c_2 \quad u_{max} = u_t + c_1|u_t| + c_2$$

$$v_{min} - v_t + c_1|v_t| - c_2 \quad v_{max} = v_t + c_1|v_t| + c_2$$

c_1 and c_2 are predetermined constants.

Determination of Person Window

The size of the person window is equal to the object window if the ratio of the height to the width of the object window is similar to the previously remembered ratio.

If the ratio of the object window is different from the remembered ratio, the person window is determined by modifying the object window based on the remembered ratio. The size of the person window is determined by the following equations:

$$width_p = width_o$$

$height_p = height_o$

$height_p = height_0$
$width_p = height_p / ratio$

where

subscript *p* is the person window and *o* is the object window.

Equation (22) is used when the difference of the width between the object window and the predicted person window is less than the difference of the height. Otherwise, equation (23) is used.

The location of the person window is equal to the object window. This case means that a part of the body of the person is occluded, or that the velocity of the part of the body is different from the velocity of the other parts of the body.

If the velocity (u_t, v_t) is similar to the background velocity (u_t^b, v_t^b), the person window is not changed and is determined by the predicted person window. In this case, the object window cannot be extracted because there is only a small difference between the background and the person on the velocity field; thus, the person stops against the camera motion.

Camera Control

The active camera head is controlled to keep the person in the field of view of the camera. Let (x, y) and (u, v) denote the center point and the velocity of the person window, respectively. For example, if the person is in the left half of the image L and moves to the left by the velocity u, the camera is turned to the left by $|u|$ pixels. The active camera head is controlled as follows:

- If $(x, y) \in L$ and $u > 0$, turn left by $|u|$ pixels.
- If $(x, y) \in R$ and $u < 0$, turn right by $|u|$ pixels.
- If $(x, y) \in U$ and $v > 0$, turn upward by $|v|$ pixels.
- If $(x, y) \in D$ and $v < 0$, turn downward by $|v|$ pixels.

where

L is the left half of the image.

R is the right half of the image.

U is the upper half of the image.

D is the lower half of the image.

Implementation

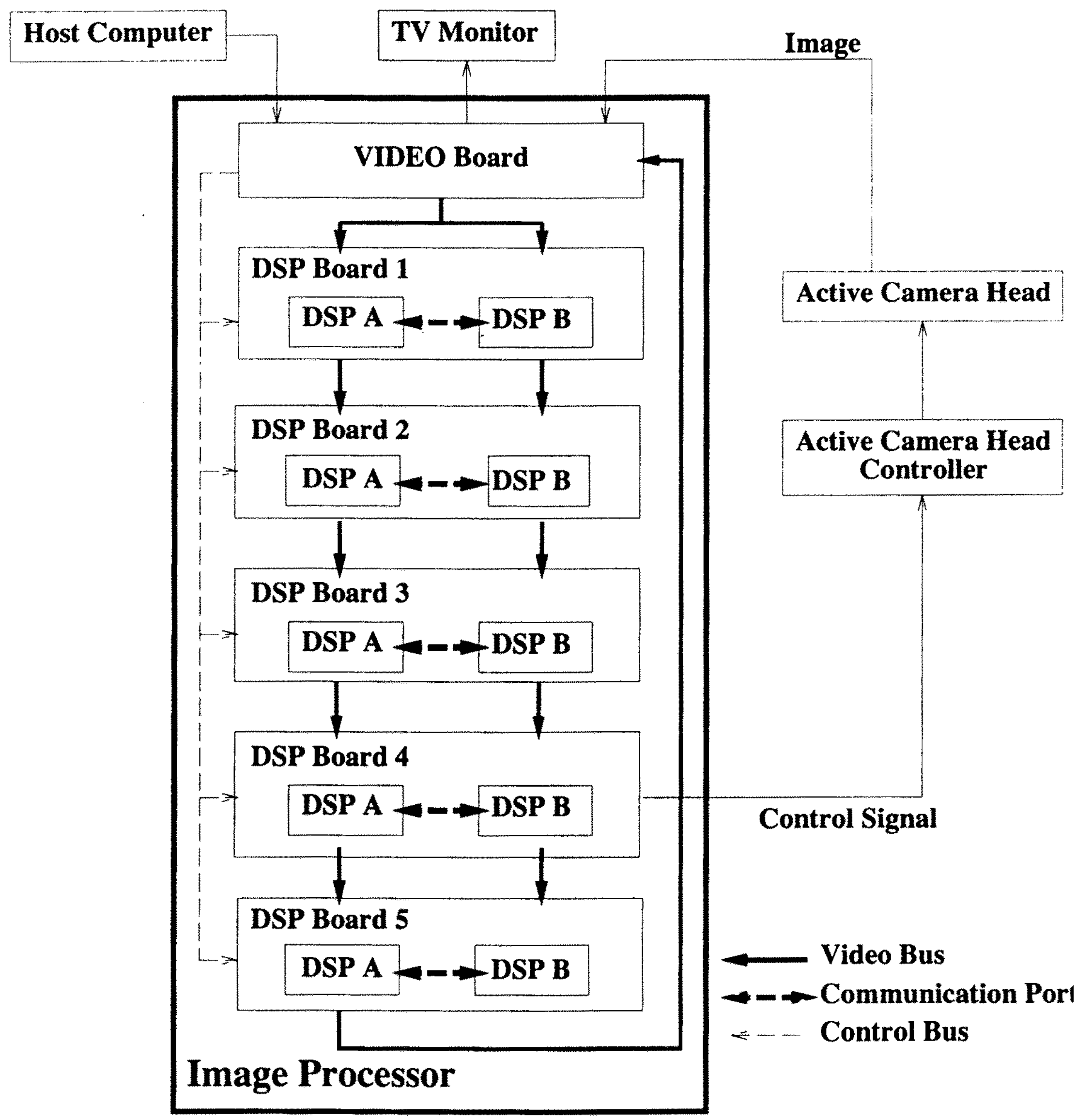
Hardware Architecture

Figure 4 shows the overview of the real-time person tracking system. The image processor has a video board and five DSP boards connected in tandem. The image processor is pipeline architecture.

The video board is the interface with the active camera head, the host computer, and the TV monitor. The DSP board can process the input image in parallel because it has two independent DSPs. We call these DSP A and DSP B. DSP A in the fourth DSP board sends the control signal to the active camera head controller. The controller turns the active camera head based on the signal from the DSP. The image processor includes the following special features.

- As the pipeline architecture is used, a burden is dispersed by increasing the DSP boards accordingly.
- Various image processings are available by changing the programs downloaded to the DSPs.
- Parallel image processing is possible because each DSP board has two independent processing sections (each section has a DSP).
- Two DSPs on the same DSP board communicate with each other through the communication ports.

Figure 4. System Overview



The video board performs three functions:

- AID transformation of the image from the active camera head
- Interface to the host computer
- Digital-to-analog (D/A) transformation of the image processed by DSP boards

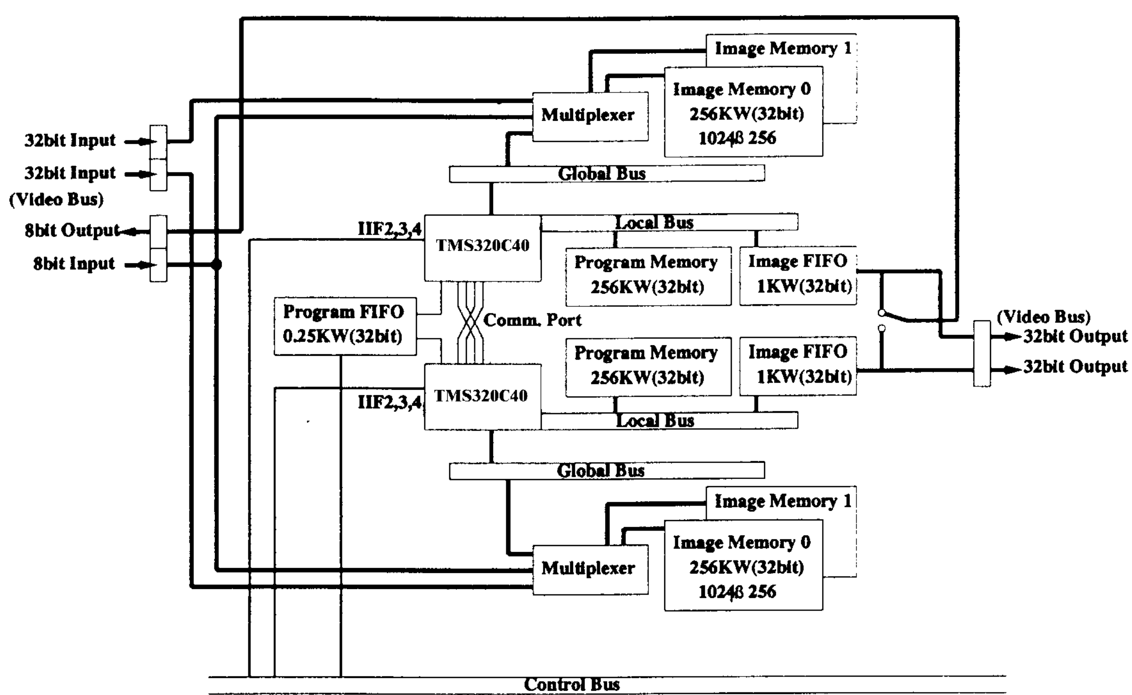
The input signal is separated to synchronous signals and image data. The image signals are digitized to 8 bit image data. The synchronous signals and the digitized image data are sent to the DSP boards through the video bus. The video board sends the interrupt signal from the host computer to the DSP boards. The video board downloads the programs from the host computer to the each DSP through the control bus.

Figure 5 shows the block diagram of the DSP board. The input data and output data of the DSP board are 8 bit or 32 bit. The 8 bit data are used for the input and output for the video board. The 8bit input data video board is written to both the image memory of DSP A and DSP B. Either DSP A or DSP B can output 8 bit data.

The 32 bit data are used for the input or the output for the DSP board. The input data is written to the image memory. Each DSP processes a image stored in the image memory according to the program downloaded by the host computer. The program memory stores the program and various data.

The result of the processing is output to the next board. This image memory has a double buffer, and the DSP can not read data from the buffer receiving the data from the previous board. The DSP board can receive data from the previous board even if the DSP is performing the image processing.

Figure 5. TMS320C40 DSP Board Block Diagram



Implementation of the Algorithm

The processing described in the *Filtering* section is assigned to the DSPs.

DSP board 1 The recursive spatial filter is convolved to the input image. DSP A convolves filter *g*. Filter *h* is convolved by DSP B.

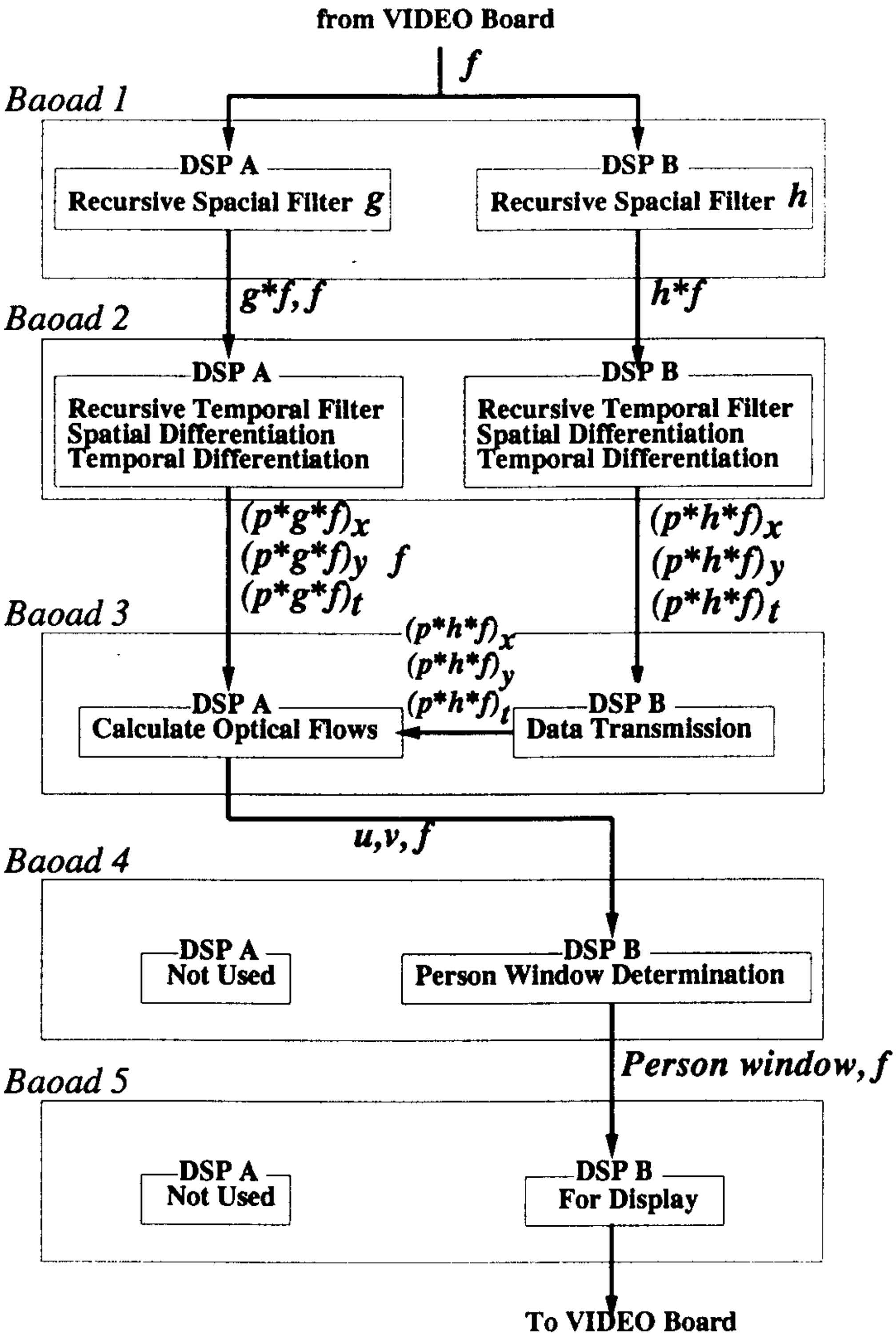
DSP board 2 DSP A and B perform the same processing. First, the recursive temporal filter is applied to the output from DSP board 1 using the result of the previous frame. And next, the differentiations with respect to the *x*, *y*, *t* are calculated. The output of the temporal filter is stored to use the next frame in the program memory.

DSP board 3 DSP B transmits the data from previous the DSP board to DSP A. DSP A calculates the optical flows based on equations (6) and (7).

DSP board 4 DSP B determines the person window based on the optical flows and the knowledge of the shape of the person. DSP A is not used.

DSP board 5 DSP B overwrites the person window on input image *f* and send it to the video board. DSP A is not used.

Figure 6. Implementation



Result of Tracking

Figure 7 shows the result of the real-time person tracking. A person is walking from left to right. The box on the floor occludes his feet. Unless knowledge of the shape of the person is used, the feet of the person can not be tracked. This method allows the system to track his whole body even though a box occludes the feet. The active camera head is controlled to keep the person in its field of view.

Figure 7. Tracking Result

Summary

We develop a real-time person tracking system. The system comprises the high-speed image processor and the active camera head. The image processor processes images in real time using TI TMS320C40 DSPs. The region of the person in the image is detected based on the optical flows and knowledge of the shape of the person. The active camera head is controlled to track the person. An experimental result shows the effectiveness of this method.

References

[1] Y.Shirai, J.Miura, Y.Mae, M.Shiohara, H.Egawa, S.Sasaki: “Moving Object Perception and Tracking by Use of DSP,” *Prc. CAMP*, pp.251-256 (1993).

[2] J.L.Barron, D.J.Fleet, S.S.Beauchemin. “Systems and Experiment Performance of Optical Flow Techniques,” *International Journal of Computer Vision*, Vol.12, No.1, pp~43-77, (1994).

[3] H.Inoue, T.Tachikawa, M.Inaba: “Robot Vision System with a Correlation Chip for Real-time Tracking, Optical Flow and Depth Map Generation,” *Proc. IEEE International Conference on Robotics and Automation*, pp.162 1-1626 (1992).

[4] M.V.Srinivasan: “Generalized Gradient Schemes for the Measurement of Two-Dimensional Image Motion,” *Biological Cybernetics*, Vol.63, pp.421-431(1990).

[5] P.Sobey, M.V.Srinivasan: “Measurement of Optical Flow Using a Generalized Gradient Scheme,” *Journal of the Optical Society of America*, pp.1488-1498 (1991).

[6] R.Derishe: “Fast Algorithms for Low-Level Vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol.12, pp.78-87 (1990).