以強度投影的適應性視窗法搜尋和追蹤目標物

研究生：孫瑗君　　　　　　　　　　指導教授：成維華

國立交通大學機械工程研究所

摘要

本文提出一個有效率的〝Adaptive window method〞來尋找和追蹤目標物。

本文中，由 CCD camera 對場景做影像擷取的工作和一部個人電腦做影像分析所組成。隨著電腦的快速發展，電腦使我們能即時地分析並做處理。在處理搜尋或是追蹤目標物時，最常用又簡單的方法就是 Correlation-based method，但此法的計算量相當的大，因此本文主要是根據 S-I. Chien and S-H. Sung 兩人所提出的〝Adaptive window method〞做了修改，目的即為了縮小搜尋的範圍以減少計算量並加快 Adaptive window 收斂的速度。

我們利用 projection-based method 代替 correlation-based method 來實現 Adaptive window，主要的理由是投影是一種統計的量，有利於去除雜訊，降低複雜的背影造成的影響。另外，投影的演算法所需的計算量很小，有利於加速整個追蹤過程。

實驗是以各種人工的真實影像序列來實現本文所提出的演算法，結果顯示此法可適用於多種含複雜背景及雜訊的影像序列，而符合追蹤的目的。

關鍵字：適應性視窗，目標物追蹤

Adaptive Window Method with Intensity Projection for Target Searching and Tracking

Student: Yuan-Hsun Sun Advisor: Dr. Wei-Hua Chieng

Institute of Mechanical Engineering

National Chiao Tung University

Abstract

According to *S-I. Chien* and *S-H. Sung* proposed adaptive window method, an efficient algorithm is proposed for the adaptive sizing of a tracking window. In this thesis, the composition of tracking is a CCD camera which collects sequence images and a person computer which analyze image. With the development of a computer quickly, we can analyze and handle instantly in computer. In tracking or following, a common and efficient method is proposed. Correlation-based method is simple, but it must have a large amount of calculation. For the reason, we modify adaptive window method to fit in with tracking. The purpose is in order to reduce calculation and speed up convergence.

We propose projection-based method instead of correlation-based method. Projection is a statistical quantity that could reduce the noises. We could achieve benefit of minimizing the influence of complex background and clutters in tracking process. In addition, projection algorithm has less computation complexity that makes the tracking faster.

Experimental results using various artificial and real sequences confirm that the proposed algorithm can effectively adjust a tracking window to a moving target and is robust to a complex background and noise.

Keywords: adaptive window, target tracking

Acknowledgement

First, I would appreciate my adviser, Professor Wei-Hua Chieng. He guides me to the right way in my research during two years. Besides, I am grateful to my seniors specially, one is ABD Chun-Pin Hsu, who teaches me how to analyze a problem and point out my mistakes for correction; another is Gen-Wei Hu, who gives me lots help in my schoolwork and programming language. I also thank my peers, Kevin, Blackman, Leader Zheng and Brother Qun, and a senior, Gley, who give me lots of help whatever in all respects.

Secondly, I thank many friends who are not in Chiao Tung University. They encourage me at any time spiritually. I cannot finish the graduated studies without them. I cannot count all of them completely, but I must say, “All my friends, Thanks a lot.”

Finally, I would like to appreciate my family, who always support me and encourage me all the time. My parents give me a way to learn and create my own life happily and luckily. I dedicate this thesis to them sincerely.

In my school life, I acquire not only all kinds of knowledge but also most precious treasure in getting along with people.

Content

List of Figures

List of Table

Chapter 1 Introduction

Artificial intelligence is a popular subject. This subject applies to computer vision is concerned with designing systems that are intelligent and with studying computational aspects of intelligence. Artificial intelligence is used to analyze scenes by computing a symbolic representation of the scene contents after the images have been processed to obtain features.

Automated tracking of objects through a sequence of images has remained one of the difficult problems in computer vision. A sequence of images must be preprocessing by image processing. Image processing techniques usually transform images into other images. On the other hand, computer vision algorithms take images as inputs but produce other types of outputs, such as representations for the object features in an image.

For this reason, the technology of image processing has been developed for a long time.

As computers get faster, more and more complex calculation and problems in image analysis not only become practically solvable in a short time, but also have the great performance in multimedia. We regard the image processing in computer as computer vision. In the word, we hope the computer can see like a human being. Therefore, the computer needs to have eyes and retinas, like a CCD camera and a frame grabber. On the occasion, the computer vision system has been established. Because of the development of these equipments, the application of computer vision system is acquired more and more easily. The CCD camera has the ability to receive images that high quality and high resolution. The frame grabber can capture images immediately to attain our requirement. Moreover, the computer hardware makes calculation more efficient. At this time, we can implement image processing by PC-base software to yield the result with efficiency.

Many fields are related to computer vision, for instance, it can be useful in traffic control in order to increase safety and obtain road state information of monitored areas. As we will see, techniques developed from many areas such as image processing, computer graphics and pattern recognition are used for recovering information from images. The goal of computer vision is to make useful decisions about real physical objects and scenes based on sensed images.

1.1 Target Tracking

It is well known that significant amount of useful information can be obtained about 3D motion and structure using the projected 2D image motion sequence resulting from the movement of a camera and that of 3D objects. In a variety of industrial and military applications, it is required to track moving objects or a scene of interest in an image sequence. Being able to track image structures over time is a useful and sometimes necessary capability for vision systems intended to interact with a dynamic world.

There is an extensive literature on tracking methods operating without specific a priori knowledge about the world such as object models or highly restricted domains. The tracking in this direction can be accomplished by mainly three methods: correlation-based tracking [1-4], optical flow based tracking [5-8], and feature-based tracking [9-12], for example, by following prominent features in a scene or by using pixel-based correlation. Specially, the correlation-based tracking acquires a reference image from the previous frame and looks for the most similar area to it from the current frame image. Optical flow techniques are based on the idea that for most points in the image, neighboring points have approximately the same brightness. In other words, the world is made up of continuous objects over which brightness varies smoothly.

In this thesis, we use adaptive window method with a feature of intensity projection to find the position, size of the object. A concept of an adaptive window could be found also in stereo matching. In the case of stereo matching, a concept of an adaptive window could be found also in stereo matching, In the case of stereo matching, the disparity boundaries become sharp for the smaller window but the computed disparity becomes sharp for the smaller window but the computed disparity becomes usually noisy. The larger window means that the computed disparity becomes clean but the disparity boundaries can be blurred.

1.2 Literature Review of Target Tracking

Feature tracking methods have an overview[13]. It gives a classification of some of the feature tracking methods in Figure 1-1. The classification of some of feature tracking methods: *Gambotto, J.-P.* [14] proposed line segments approach to solve the feature tracking problem. Using straight line correspondences, one can determine the motion parameters as well as structure information of the objects. Line tracking may fail in situations which give rise to the so-called “aperture problem”; Block matching is another approach to tackle the feature tracking problem. *El-Azim, S.A., Ismail, I.,*and *El-Latiff, H.A.;*[15] proposed motion estimation using block matching. The resultant motion vectors obtained by BMA are scanned to count the repetitions for each motion vector; Curves and contours can also be used for tracking [16]. Tracking contours and curves is particularly useful in the absence of any geometric model of the observed structure. *Tissainayagam,P.* and *Suter, D.* made the tracker which is coupled with an automatic motion-model switching algorithm robust and reliable when the object of interest is moving with multiple motion. The model-based tracking technique provided is capable of tracking rigid and non-rigid object contours with good tracking accuracy.

There are several algorithms in the literature for tracking point features. *Mehrotra, Rajiv*[17]proposed a new polynomial time algorithm for finding feature point trajectories from a large sequence of frames. According to the constraints of the paper, the changes in the motion characteristics of a point are small over a short period of time and are temporally continuous. This implies that the trajectory of a point over a given time interval is as “smooth” and “short” as possible.

1.3 Thesis Retrospection

First, I appreciate these thesis writers very much. I will list authors’ names and subjects according to date.

Chun-Pin Hsu [18], “The Transmission and Control of a Remote Image Based on File Transfer Methods”.

W. H. Tam [19], “A Human Face Tracker System Design Study Using the Technology of Digital Image Process”.

Jin-Hong Li [20], “Image Based Digital Monitoring System”.

Jui-Hsiung Peng [21], “Moving Frame Based Method to the Inspection of Assembled PCB”.

Ya-Wen Chiou [22], “Motion Pursuit in Active Vision Systems”.

Kuo-Feng Chang [23], “Image Processing for Multiple Moving Objects in Surveillance and Recognition”.

Gen-Wei Hu[24], “Automatic Image Detection System for Human Shape Analysis”.

1.4 Objective

In the thesis, a concept of adaptive window method is from stereo matching and feature tracking of intensity projection. First, we select a target in the first image of a sequence of images. Generally a human operator establishes an initial window with its size as a search region. Then, the position and size of windows are updated by adaptive window algorithm. We enclose the target tightly with our proposed adaptive window method and could achieve benefit of minimizing the influence of complex background and clutters in tracking process. Finally, we could estimate the state of the target by methods of image processing and pattern recognition efficiently and reliably. We make a CCD camera have a pair of eyes like a human and detect the moving object.

1.5 Thesis Structure

This thesis has six chapters. Chapter 1 is introduction. Chapter 2 describes the scenario and block diagram of target tracking with adaptive window method, and the object movement. Chapter 3 introduces the method of adaptive window algorithm with intensity projection for target tracking. We propose the implementation of the adaptive window method in Chapter 4. Chapter 5 is the experiment and results. Finally, we make conclusions and future development in chapter 6.

Chapter 2　Overall architecture of target tracking

The target tracking from a surveillance area is the major process in image processing based analysis techniques. The most common and simple algorithm used for target tracking is based on background differencing and feature operati. In the section 2.1, we introduce scenario of vehicle-following. Then, section 2.2 describes the block diagram of this system. Final, section 2.3 illustrates the fundamental movement of an object.

2.1 Scenario of Target Tracking

The scenario of target tracking is shown in Figure 2-1. The environment of the system may be indoor or outdoor. The system can capture the moving object which we interest in attentive area. The appearance hardware of target tracking uses a single Charge-Couple Device (CCD) camera, a personal computer, and a monitor. A CCD camera can take a sequence of image with enough sampling time.

The CCD camera acquires the fixed area which we want to keep under surveillance. A sequence of images is delivered to our personal computer, and we process a sequence of image in accordance with the CCD camera. Useful information about what the image contains can be obtained by analyzing and understanding the difference between images caused by the motion. The functions of a personal computer are receiving a sequence of images from the

CCD, executing the adaptive window algorithm and computing all difficult processing of math. In addition, the monitor is the interface of the tracking system. The surveillance area, moving objects, and control buttons are shown.

2.2 Block Diagram of the Detection System

Block diagram of target tracking is showed in Figure 2-2. The image acquisition gets images through the static CCD camera, delivers to personal computer and shows them on the screen.

For image processing applications, the sequence of images have to be processed at a rate of enough sampling time which produces a large amount of data. Therefore, the image processing algorithms have to be simple but effective so that they can be executed in real-time. For the purpose, the first image processing algorithm required to shrink the working region where the object target appears.

Before motion of the targets, the preprocessing is necessary, because many applications require obtaining the most quality digital image. First, we can get the first incoming image with a moving object and background. We select the moving target which we want to track and enclose it with a rectangle. Then, we record all information which we want and can use in adaptive window method, such as grey-level values of pixels, the coordinate position of the target or enclosed rectangle in whole image, and so on. Finally, according to the algorithm of target tracking in later chapters, we can process a sequence of new incoming image and distinguish from the former image. Thus, we can track the target with image processing in target tracking.

2.3 Motion of Object

We describe object movement relative to the imaging plane as translation, rotation, and scale [25]. Rotation which is an angular displacement about a specific axis includes turning left and turning right. For a three-dimensional (3D) object, rotation will make the contours and the appearance of the object change. However, we hypothesize that the camera is far away from objects, as in remote sensing, then, three-dimensionality of objects can be ignored and structure using the projected two-dimensional (2D) image motion sequence. Translation is object displacement along the image abscissa, ordinate, or both. A change in scale is an object's increase or decrease in size relative to a fixed ratio within an image. These terms are illustrated in Figure 2-3.

In this thesis, we lock our object may be a car, a helicopter, a flight, a ball and so on, anything is possible. Then, no matter the target has translation, rotation, or scale we must draw a rectangle to enclose it.

Chapter 3 Adaptive Tracking Window Algorithm Analysis

According to adaptive window method with sizing vectors which *S-I. Chien* and *S-H. Sung* bring up [2] [26], we propose another more simple and efficient adaptive window method with intensity projection that can provide a tracker with a tight reference window by adaptively adjusting its window size and position independently into four side directions.

The architecture of the system will be introduced. In the section 3.1, we introduce some basic operators which are used in image processing. Preprocessing in incoming image sequence will be listed in section 3.2. We expatiate on the algorithm in the last section of the chapter. Further, we will describe the implementation of the algorithm completely in next chapter.

3.1 Basic Image Operator

When receiving a sequence of images from CCD, the image quality is not good. In image processing, we need the better quality and high-resolution of image source. The preprocessing is done after the frame grabs the image. In this chapter, we introduce all basic image operations that we use in adaptive window method in this section.

3.1.1 Image Smoothing Operator

When the image is acquired from the CCD camera, there is always noise, which is usually caused by the transmission cable and the contact, in the image. The noise will influence the result of certain process even make this system failure.

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by filtering [27].

The filtering operation can be implemented by convolving the entire image with a sample 3 × 3 or 5 × 5 mask. This method of the low-pass filter is used in image smoothing.

Low-pass Spatial Filter

The output of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called averaging filters.

The spatial structure of a 3 × 3 low-pass filter $LP_{3\times3}$ is represented as Eq. (3-1):

$$LP_{3\times3} = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{3-1}$$

Use of the first filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask into Eq. (3-2):

$$R = \frac{1}{9} \sum_{i=1}^{9} z_i \tag{3-2}$$

Where R is the result (or response) of linear filtering with a filter mask at a point (x, y) in the image, and the z's are the values of the image gray levels corresponding to those coefficients. Eq. (3-2) is the average of the gray levels of the pixels in the 3 × 3 neighborhood defined by the mask. Note that, instead of being 1/9, the coefficients of the filter are all 1's. The idea here is that it is computationally more efficient to have coefficients valued 1.

3.1.2 Image Sharpening Operator

The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition, such as edges detection. Sobel masks mainly for detecting diagonal edges.

Sobel Operator

Before use PDOE operator which is introduced in next section, edge detection in the image must be made. Edge detection is by far the most common approach for detecting meaningful discontinuities in gray level. The Sobel operator is the magnitude of the gradient computation. The gradient operator, S_x and S_y, can be implemented using convolution 3 ╳ 3 mask as following Eq. (3-3):

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3-3}$$

Note that the coefficients in the masks sum to 0, indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator. This operator places an emphasis on pixels that are closer to the center of the mask. Figure 3-1 shows the performance of this operator. The Sobel operator is one of the most commonly used edge detectors.

3.1.3 PDOE Operator

Image Difference is a common operation in image processing. We can find the variation of gray-level values between two images by this operation. In general, gray-level values of the target are larger than gray-level values of the background, so we use PDOE (Positive Difference Operation of Edge) [2] [26] to find the location of the target.

We represent PDOE image of between two images as $PDOE(x,y,k)$

$$D(x,y,k) = E(x,y,k) - E(x,y,k-1) \quad (3\text{-}3)$$

$$PDOE(x,y,k) = \begin{cases} D(x,y,k) & , \text{ if } D(x,y,k) > 0 \\ 0 & , \text{ otherwise} \end{cases} \quad (3\text{-}4)$$

Where $E(x,y,k)$ represents an edge component of a point (x,y), and $E(x,y,k-1)$ is obtained from the previous frame. Note that the edge component is acquired by the Sobel operator.

The PDOE operator removes negative components in $D(x,y,k)$ that is not common in conventional difference operation. The PDOE can extract single edge component for the target wile an absolute difference method produces double moving components for a moving target. This is useful for removing the background components and detecting motion components more stably. Figure 3-2 is an example of PDOE operation.

3.1.4 Dilation Operator

With A and B as sets in 2-D plane, the dilation process consists of obtaining the reflection of B about its origin and then shifting this reflection by x. The dilation of A by B then is the set of all x displacements such that $\hat{B}$ and A overlap by at least one nonzero element. In definition, we call B as structuring element of A or probe B. The mathematical morphological formula of dilation is written as:

$$A \oplus B = \{ x \mid [(\hat{B})_x \cap A] \subseteq A \} \tag{3-5}$$

Dilation operator is used to mend the small inner holes in the moving object and make the variance mask. Figure 3-3 is an example of dilation operation. The symbol A and B are the basic graph and structuring element.

3.2 Preprocessing

A success of target tracking largely depends on choosing suitable window size and position and thus transferring the proper reference image to the next frame. Since the adaptive algorithm is capable of adjusting a reference image size more properly, we can minimize the influence of complex background and clutter. Therefore, we must execute preprocessing before adaptive window algorithm. In the case of context of adaptive window algorithm, we also use a special measure property, intensity projection that has no influence of background and noise.

The structure of the target tracking is shown in Figure 3-4. When a sequence of image is coming, the PDOE operator is used. The feature set used for adjusting an adaptive window is the PDOE image previously described. Through the operation, we can acquire positions and sizes of moving targets. To reduce and blur noises, the low pass spatial filter is used. Finally, we use dilation operator to increase information of PDOE image and make the feature set used for adjusting an adaptive window previously.

3.3 Proposed Algorithm for Adaptive Sizing of Tracking Window

An efficient algorithm is proposed. The adaptive window algorithm needs mainly several steps: adaptive window setting, the direction and magnitude of four side vectors, and the window size determination. Here, the direction and magnitude of four side vectors are independently each other. Three steps as following sections will be introduced. Beside, the flow chart of the adaptive window algorithm is shown in Figure 3-5.

3.3.1 Adaptive Window Setting

First, we design three frames for adaptive window setting. One is adaptive reference frame ($F_M(k)$) and the others are inner frame ($F_I(k)$) and outer frame ($F_O(k)$). The layout of an inner frame and an outer frame is for adaptively controlling the reference window. We define the inner frame is shrunk and the outer frame is expanded 1/6 times by the reference frame. Since we have designed three frames, we now construct three regions: inner, middle, and outer regions in Figure 3-6 where m is the width of $F_M(k)$ and n is the height of $F_M(k)$. According to priority, the outside boundaries of the three regions are inner, reference, and outer frames.

Second, we define the outside boundary of the middle region as the adaptive window boundary given by the previous tracking stage. We design the inner region needed for extracting information within an object boundary. In addition, the extended outer region, we request the moving target must within this region on the next tracking stage. The extracted information from three regions for each direction will be used as a criterion for determining whether the object is in our tracking range or not. Finally, each direction has three regions so an adaptive window set has twelve districts. They are top-out (TO), top-middle (TM), top-in (TI), bottom-out (BO), bottom-middle (BM), bottom-in (BO), left-out (LO), left-middle (LM), left-in (LI), right-out (RO), right-middle (RM) and right-in (RI) shown in Figure 3-7, where p is width of $F_I(k)$ and q is height of $F_I(k)$, and we acquire information with 50% width and height of inner frame. Therefore, p and q can be written by Eq. (3-6) and Eq. (3-7)

$$p = \text{width of } F_I(k) = m - \frac{m}{6} \times 2 = \frac{2}{3}m \tag{3-6}$$

$$q = \text{height of } F_I(k) = n - \frac{n}{6} \times 2 = \frac{2}{3}n \tag{3-7}$$

Due to the window setting, we also define the size of each region, as Eq. (3-8)-(3-11):

$$\text{TI, BI} \begin{cases} \text{width} = 50\% \text{ width of } F_I(k) \\ \text{height} = \dfrac{n}{3} \end{cases} \tag{3-8}$$

$$\text{TM, TO, BM, BO}\begin{cases}\text{width} = 50\% \text{ width of } F_I(k)\\ \text{height} = \dfrac{n}{6}\end{cases} \tag{3-9}$$

$$\text{RI, LI}\begin{cases}\text{width} = \dfrac{m}{3}\\ \text{height} = 50\% \text{ height of } F_I(k)\end{cases} \tag{3-10}$$

$$\text{RM, RO, LM, LO}\begin{cases}\text{width} = \dfrac{m}{6})\\ \text{height} = 50\% \text{ height of } F_I(k)\end{cases} \tag{3-11}$$

3.3.2 The Direction and Magnitude of Four Side Vectors

In Figure 3-8, there are four reference unit vectors for four sides. We define the four reference unit vectors: u^i are four unit edge vectors and $i = 0,1,2,3$ represent four directions. The strategies of expanding or shrinking are shown as Eq. (3-8) and Eq. (3-9)

Expanding: $B^i = \omega^i \cdot u^i$ (3-12)

Shrinking: $B^i = -\omega^i \cdot u^i$ (3-13)

Where i = 0, 1, 2, 3 represent right, top, left, bottom direction, B for horizontal direction, i.e., i = 0, 2, left and right of a window is given, on the other hand, i = 1, 3 represent vertical direction. The B's are side vectors, the u's are the directions of unit vectors and the ω's are the magnitudes of side vectors. Before we determine ω, we must calculate the intensity projections of the twelve districts include the four directions and the three regions first as shown in Eq. (3-14) and Eq. (3-15)

$$\begin{cases} proj_{RGN}{}^{i}(y), & i = 1,3 \\ proj_{RGN}{}^{i}(x), & i = 0,2 \end{cases} \tag{3-14}$$

$$\begin{cases} proj(y) = \sum_{x=LB}^{RB} I(x,y) \\ proj(x) = \sum_{y=BB}^{TB} I(x,y) \end{cases} \quad (3\text{-}15)$$

where RGN represents I, M, O, i.e. the inner, middle, outer regions. $I(x,y)$ represents intensity of the PDOE image. The $proj$'s are the intensities of pixel counts along x or y axis whose range as their width or height of the region. In Eq. (3-15), RB, LB, TB ,and BB represent the right boundary, left boundary, top boundary, and bottom boundary of a certain region. Horizontal and vertical projections can be easily obtained by finding pixels for each bin in the vertical and horizontal direction. Figure 3-9 represents the projection analysis in twelve regions.

There is another problem needs to be solved. In the same direction, how we select suitable one of them, and that can make the tracking result better. We set two principles to determine which one we select suitably. One is choosing the maximum projection for each direction as Eq. (3-16)-(3-17)

$$PROJ_{RGN}{}^{i} = \max\{proj_{RGN}{}^{i}(y)\} \qquad i = 1,3 \quad (3\text{-}16)$$

$$PROJ_{RGN}{}^{i} = \max\{proj_{RGN}{}^{i}(x)\} \qquad i = 0,2 \quad (3\text{-}17)$$

Where RGN mean I, M, O, and Eq. (3-18)-(3-19) show that the value, x' and y', when the maximum projection occur.

$$y'_{RGN}{}^{i} = \arg(\max\{proj_{RGN}{}^{i}(y)\}) \qquad i = 1,3 \quad (3\text{-}18)$$

$$x'_{RGN}{}^{i} = \arg(\max\{proj_{RGN}{}^{i}(x)\}) \qquad i = 0,2 \quad (3\text{-}19)$$

When $\max\{w_I \times PROJ_I,\ PROJ_M,\ PROJ_O\}$ occur, one side of the adaptive window can be determined as Eq. (3-20)-(3-21). Note that we set a weight to inner projection information, because the surface of targets has much edge information that causes too much inner information. We must reduce the inner information, so we define $0 < w_I < 1$. For various targets, we can set the most adaptive weight value. Generally, $w_I = \frac{1}{4}$, the result is from our experiment for a incoming sequence of image without no noise.

$$y'^{\,i}_{RGN} = \arg(\max\{w_I \times PROJ_I,\ \ PROJ_M,\ \ PROJ_O\}) \qquad i = 1,3 \qquad (3\text{-}20)$$

$$x'^{\,i}_{RGN} = \arg(\max\{w_I \times PROJ_I,\ \ PROJ_M,\ \ PROJ_O\}) \qquad i = 0,2 \qquad (3\text{-}21)$$

However, when $PROJ_I$, $PROJ_M$ and $PROJ_O$ are similar to each other, the maximum numbers of the pixels are used in place of the second policy, the summation of gray levels, as shown in Figure 3-10. Hence, the other principle is choosing the maximum amount. That can provide more correct edge information and avoid bringing the error caused by little bright noises. We compare with the variation between the maximum and the second maximum values in the same direction. If the variation is closed we will take the second policy as following,

$$\text{first\_}\omega^i = \max\{w_I \times PROJ_I,\ \ PROJ_M,\ \ PROJ_O\}$$

$$= \text{the max value in the same direction} \qquad (3\text{-}20)$$

$$\text{sec\_}\omega^i = \text{sec\_max}\{w_I \times PROJ_I,\ \ PROJ_M,\ \ PROJ_O\}$$

$$= \text{the second max value in the same direction} \qquad (3\text{-}21)$$

Then, if the variation between $\text{first\_}\omega^i$ and $\text{sec\_}\omega^i$ with standard derivation is less then 200, we figure out the numbers of pixels where $\text{first\_}\omega^i$ and $\text{sec\_}\omega^i$ occurs. We will take the maximum numbers strategy to improve the first policy. We compute $n_{\text{first\_}\omega}$ and $n_{\text{sec\_}\omega}$. $n_{\text{first\_}\omega}$ represents numbers of edge pixels in y' or x' and n_{sec_ω} represents numbers of edge pixels in y' or x''. The maximum numbers policy is shown in Eq. (3-22)

$$\omega = \max\{n_{first\_\omega}, n_{sec\_\omega}\} \quad (3\text{-}22)$$

And

$$y', x', y'', \text{or } x'' = \arg(\max\{\omega\}) \quad (3\text{-}23)$$

Thus, we acquire the magnitude of the side vectors. In next section, we have a complete inference of the position and size of a new window.

3.3.3 The Reference Window Relocation with Side Vectors

Above sections, the magnitudes and directions are determined, the final window size is also determined. We design the directions of side vectors with shrinking and expanding. ω^i occur in the following three regions: inner region, middle region and outer region. If ω^i is in inner region or middle region represents shrinking adaptive window. Otherwise, ω^i is in outer region represents expanding adaptive window. Thus, we can determine the final magnitude of adaptive window. Eq. (3-24)-(3-26) is shown for vertical direction ($i = 1, 3$) and Eq, (3-27)-(3-29) is shown for horizontal direction ($i = 0, 2$)

For vertical direction:

Inner: $$\Delta B^i = |y - y'^i| \quad or \quad |y - y''^i| \quad (3\text{-}24)$$

Middle: $$\Delta B^i = |y - y'^i| \quad or \quad |y - y''^i| \quad (3\text{-}25)$$

Outer: $$\Delta B^i = |y - y'^i| + \frac{n}{6} \quad or \quad |y - y''^i| + \frac{n}{6} \quad (3\text{-}26)$$

For horizontal direction:

Inner: $$\Delta B^i = |x - x'^i| \quad or \quad |x - x''^i| \quad (3\text{-}27)$$

Middle: $\Delta B^i = \left|x - x'^i\right| \quad or \quad \left|x - x''^i\right|$ (3-28)

Outer: $\Delta B^i = \left|x - x'^i\right| + \frac{m}{6} \quad or \quad \left|x - x''^i\right| + \frac{m}{6}$ (3-29)

Where ΔB represents the displacement of the side vectors and is determined by $y', x', y'',$ or x''. Finally, we know relocate the position of the boundaries according to the change of window size p^i, for $i = 0, 1, 2, 3$, the origin of coordinates can be simply given by Eq. (3-30)

$$\tilde{p}^i = p^i + \text{sgn} \cdot \Delta B^i \tag{3-30}$$

Where

$$\text{sgn} = \begin{cases} +1 & \omega \text{ is in outer region} \\ -1 & \omega \text{ is in inner or middle region} \end{cases} \tag{3-31}$$

And p^i is the coordinate value of the previous window position before the window sizing procedure. Figure 3-11 illustrates such relocation of the tracking window by Eq. (3-30).

Chapter 4 Implementation of Adaptive Window Method

In former chapter, we introduce the adaptive window algorithm. We will analyze the complete flow path to realize the adaptive window method. In section 4.1, the implementation steps of adaptive window algorithm are proposed. By the way, we provide the illustration of the method in section 4.2.

4.1 Proposed Algorithm for Adaptive Sizing of Tracking Window

Before we carry out our adaptive window algorithm, we must make the target separate from the background. Furthermore, the incoming images pass through a preprocessing with PDOE, low pass, and dilation operators. We detect edges of two images, and execute PDOE operator in our incoming images. We acquire an edge gray level image, and make PDOE image through a filter to eliminate noises. Then, we use dilation operator for increasing the information of the target.

The complete flow is in Figure 3-5. First, we find out what we want to track. Next, accomplish the adaptive window algorithm. Finally, update the new pattern and keep on these steps in turn. The steps are as following:

Step 1: frame initial window

In the beginning of tracking, generally, a human operator established an initial window with its size much larger than the object size

Step 2: set adaptive window

According to section 3.3.1, the adaptive window setting method of the inner frame and outer frame is the second step. It is shown in Figure 3-6.

Step 3: define the detective or operating region

The same as step2, the definition of twelve regions is proposed in section 3.3.1. It is shown in Figure 3-7.

Step 4: make sure the target exists

The step ensures the target is detectable and exists on our searching area. If there are no targets in our framed searching area, return to the step2.

Step 5: adjust the weight of inner region

For three regions, there are three weights for inner, middle, and outer regions individually. We usually only adjust w_I. The size of the inner region is twice than the other regions, so we design the weights as $w_I = \frac{1}{2}$, initially. However, an object has many edges in the inner part, so we make w_I smaller fairly unless the noises are more than the information which we can acquire. According to our experiment, we give a weight adjustment results in Table 5-1. We figure out a general weight value is $\frac{1}{4}$

Step 6: calculate the magnitude of the side vectors

We find out the projection with pixel intensity for four directions. We conclude two principles to decide the values of ω's. One is the maximum projection; another is the maximum numbers of one pixel. Then, we will acquire the magnitude of size vectors.

Step 7: point out the directions of the side vectors

By the region where ω's occurs, we know the target expanding or shrinking. Thus, the directions of the side vectors are decided. There are two situations of the directions, positive or negative. If ω occurs in the outer region the direction of a side vector is positive, i.e. the side is outside of the previous window. Contrariwise, if ω occurs in the inner or middle regions, the side is inside of the previous window.

Step 8: update the reference window with side vectors

Since we acquire the magnitude and directions for each side vector in step 5 and step 6, the new position and size of a new adaptive window are formed. The final side vectors are decided. It is shown in Figure 3-11.

Finish the above steps, we acquire a new adaptive window and review these steps get the new state of the vehicle.

4.2 Illustration of Implementation

Here, we selected two image sequences with a car going through a way and get across a road junction. For the first image sequence of a car which are going through a way with 35 frames. Figure 4-1 represents part of the test image sequence which is mainly translation and scale. For another example for image sequence of a car which are getting across a road junction with 45 frames. This part of image sequence has three behaviors: translation, scale, and rotation are shown in Figure 4-2.

Chapter 5 Experiment Result

We have applied our proposed method based on four independent side vectors to several image sequences and obtained the results. In this chapter the experiment of adaptive window method will be discussed. In accordance with three situations, we do some experiments respectively. In section 5.1, hardware and software which we use are proposed. The implementation of tracking with adaptive window method is in section 5.2. Tracking example of the proposed method for an artificial image sequence with uniform noises brought by the software of image processing. Finally, we take a special example in final section.

As a result of our experiment, we can execute the adaptive window method in a frame that needs 60 to 100 ms. The average time needs about 80 ms; on the other hand, 11 or 12 frame can be complete in one second averagely. The proposed method is fast enough and achieves the condition of real-time tracking.

5.1 Description of Hardware and Software

The environment of hardware is a personal computer(PC)of Intel Pentium Ⅳ 2.4 GHz with 512MB RAM. The PC receives a sequence of images from the CCD, executes the adaptive window algorithm and computes all difficult processing of math. Canon PowerShot S45 is the camera to capture the dynamic image frames. The captured images are 320(h)*240(v) pixels, 24 bits color image.

The software of Paint Shop Pro 6 is used to transfer the captured color images to 8 bits gray level images and add noises in experiment. The development environment of software is under the platform of Microsoft Windows XP. The software is programmed with Microsoft Visual C++ development and Microsoft Foundation Class (MFC) Library that provide a convenient C++ member function interface to the structures in Windows that they encapsulate. In addition, the Intelligent Mechtronical Laboratory provides the HIL-Image function class, includes basic image process and transformation.

5.2 The Implementation of Tracking with Adaptive Window Method

In chapter 4, we proposed the implementation of adaptive window method with two image sequences and it is shown in Figure 4-1 and Figure 4-2. We give the other example for different image sequence with more than one hundred frames in Figure 5-1. There is a car coming from far distance. The car is a little larger and larger, and then, it disappears from the screen of the CCD. In Figure5-1, whatever the car is translation, scale or rotation, we can frame the target car an adaptive window.

We refer *S-I. Chien* and *S-H. Sung* bring up [2] [26] to quantify the window size error E_s as follows:

$$E_s = \frac{\left|A - \tilde{A}\right|}{\tilde{A}} \tag{5-1}$$

where A is the area of the adjusted tracking window and $\tilde{A}$ is the actual area of the known target. Error evaluation, Figure 5-2, represents the window size error for the example of Figure 5-1. In the beginning, we give a bigger searching area to frame our moving target. Several frames about three to five pass through, we can get a moderate adaptive window. The initial tracking window size is set to 142×142 pixels much larger than the moving target. The size of the moving target changes slowly from the initial tracking window.

5.3 The Test of an Image Sequence with Noise

Image-sequence analysis for real-time applications requires high quality and highly efficient algorithms for tracking as there is no time to do the costly object recognition each time a new image is captured. Therefore, in order to reduce the complicated background and noise, we use a quantity which has a property that has smaller or no influence of background and noise, that is, intensity projections.

For the same image sequence with different percentages of uniform noise, we do an experiment. We add the different percentages of uniform noise with the develop software, Paint Shop pro 6. The square boundary image is our target is shown in Figure 5-3. Figure 5-3 is the experiment result with the square boundary image sequence has no noise, a clear background. There is another experiment result is shown in Figure 5-4 with the image sequence has 15 percent uniform noise. Figure 5-5 has the image

According to Eq. (5-1), we acquire three pieces of the window size error evaluation chart is shown in Figure 5-6, respectively. We combine three curves of Figure 5-6 in Figure 5-7. In Figure 5-7, compared with the three curves, there is only small influence of the noise, as long as the edge of the target can be distinguished.

5.4 Two Special Examples for Going Across

There are several moving objects which are smaller then our target near or across by our target. In Figure 5-8, the vehicle is our target and there are several bikes go across around it. We can see the adaptive window is shown in frame 1. From frame 3 to frame 18, a bike pass through our target and the adaptive window framed the bike until it separates from our target. The same situation occurs from frame 35 to frame 42.

However, when two vehicles go across each other, the adaptive window will frame the bigger intensity one. That will cause tracking the other target which we don't want. In addition, the different initial position of the window will cause the different results. The two situations are shown in Figure 5-9 and Figure 5-10. We will discuss the performance in the next chapter.

5.5 The Cause Analysis of Tracking Error

The adaptive window method can be used in common situation from the experiment; however, it will cause the tracking error and the causes as following:

1. The features of the target are unobvious or invisible:

   When the features in surface of the target are unobvious or invisible, we lose some information of the target such as edges. We acquire the target information using the intensity projection with edges cause by PDOE operator. The main causes result in unobvious and invisible features are light and colors. The intensity of light is not uniform and colors of the target and the background are similar.

2. The moving object go across the target:

   When the two moving object go across each other, the tracking errors may be occur. If the moving object is much smaller than our target, the condition can be ignored. However, if the moving object is equal or bigger than our target, the adaptive window will re-elect the target which has larger intensity projection. For the reason, the tracking error occurs.

3. The target is hidden by other objects:

   The situation is similar to the above-mentioned two reasons. The target is hidden by other objects that causes the target has no features to track result in the error occurs.

Chapter 6 Conclusion

We have presented an image tracking method which can select appropriate window sizes and positions adaptively. The method mainly proposed based on *S-I. Chien* and *S-H. Sung* bring up [2] [26]. We want to make tracking faster. Due to inertia, the motion of a physical entity cannot change instantaneously. If a frame sequence is acquired at a rate such that no dramatic change takes place between two consecutive frames then for most physical objects. No abrupt change in motion can be observed. For the reason, the side vectors in adaptive window method are proposed. Compared with the tracking method which *S-I. Chien* and *S-H. Sung* proposed is shown in Table 6-1.

In this thesis, we use the statistical quantity of projection. That's the reason why we use projection as the determinant. Reducing the noises is the best advantage of the projection. Furthermore, the current study proposed an adaptive sizing algorithm that can adaptively change the size of a tracking window to suit a target in correlation-based method. However, for the computation complexity in correlation is $\mathrm{O}(n^2)$, but in projection is only $\mathrm{O}(n)$.

At the onset of tracking, we frame an initial window much bigger than our target. The speed of convergence is another advantage is faster. The appropriate window is formed adaptively about five frames less than the original method fifteen frames. Only a drawback needs improve, that is the projection algorithm makes tracking more efficient but less accuracy

The development in the future, we can increase the accuracy first. Then, we think add the compensation of the background. Thus, the tracking system can be applied in much more situations.

Reference

[1] *Jae Gon SON; Chae Whan LIM; Il CHOI; Nam Chul KIM,* “Adaptive Sizaing of Tracking Window for Correlation-Based Video Tracking” ,IEICE TRANS.INF. & SYST., Vol.E85-D, No.6, June 2002

[2] *Chien Sung-I ; Sung Si-Hun,* “Adaptive window method with sizing vectors for reliable correlation-based target tracking” , Pattern Recognition, Volume: 33, Issue: 2, pp.237-249, Feb. 2000

[3] *Dennis A. Montera; Steven K. Rogers; Dennis W. Ruck; Mark E.Oxley,* “Object tracking through adaptive correlation” , Optical Engineering, Vol: 33, No. 1, January 1994

[4] *N.C. Kim; I.H. Jang; J.D. Kim; J.K. Kim,* “Real-time automatic target tracking based on spatio-temporal gradient method with generalized least square estimation” , Proc. IEEE ICASSP-88, vol.2, pp.1088-1091, New York, April 1988

[5] *SukHwan Lim; El Gamal, A.,* “Optical flow estimation using high frame rate sequences” , Image Processing, 2001. Proceedings. 2001 International Conference on , Volume: 2 , Pages:925 – 928, 7-10 Oct. 2001

[6]_*Yamamoto, S.; Mae, Y.; Shirai, Y.; Miura, J.,* “Real-time multiple object tracking based on optical flows” , Robotics and Automation, 1995. Proceedings.,

1995 IEEE International Conference on , Volume: 3, pp. 2328 -2333, 21-27 May 1995

[7] *Y. Huang,* "Optic Flow Field Segmentation and Motion Estimation Using a Robust Genetic Partitioning Algorithm" , IEEE Transactions on Pttern Analysis and Machine Intelligence, vol.17, NO.12, December 1995

[8] *Sengupta, K.; Wang Shiqin; Ko, C.C.; Burman, P.;* "Automatic face modeling from monocular image sequences using modified non parametric regression and an affine camera model" , Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pp.524 – 529, 28-30 March 2000

[9] *Xangdong Xie; Sudhakar, R.; Hanqi Zhuang;* "Real-time eye feature tracking from a video image sequence using Kalman filter" , Systems, Man and Cybernetics, IEEE Transactions on , Volume: 25 , Issue: 12 , pp.1568 – 1577, Dec. 1995

[10] *M.Ogata and T. Sato,* "Motion-detection model with two stages: spatio-temporal filtering and feature matching" , J. Opt. Soc. Amer., pp.337-387, 9 March 1992

[11] *Zhongxiang, L.; Liang Ronghua;* "Complex multiple features tracking algorithm in motion capture" , TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering , Volume: 1 , pp.277 – 280, 28-31 Oct. 2002

[12] *Cadman, L.; Tjahjadi, T.;* "Simultaneous feature tracking and

three-dimensional object reconstruction from an image sequence” , Image Processing, 2001. Proceedings. 2001 International Conference on, Volume: 2 , pp.391 – 394, 7-10 Oct. 2001

[13] *Srivatsan Krishnan; Daniel Raviv;* “2D Feature Tracking Algorithm for Motion Analysis” , Pattern Recognition, Vol. 28, No. 8, pp.1103-1126, 1995

[14] *Gambotto, J.-P.;* “Tracking points and line segments in image sequences” , Visual Motion, 1989.,Proceedings. Workshop on , pp.38 – 45, 20-22 March 1989

[15] *El-Azim, S.A.; Ismail, I.; El-Latiff, H.A.;* “An efficient object tracking technique using block-matching algorithm” , Radio Science Conference, 2002. (NRSC 2002). Proceedings of the Nineteenth National , pp.427 – 433, 19-21 March 2002

[16] *Tissainayagam,P.; Suter, D.;* “Contour tracking with automatic motion model switching” , Pattern Recognition, Vol.36, Issue: 10, pp.2411-2427, October, 2003

[17] Mehrota; Rajiv; “Establishing motion-based feature point correspondence” , Pattern Recognition, Volume: 31, Issue: 1, pp.23-30, 1 January 1998

[18] *Chun-Pin Hsu*, “The Transmission and Control of Remote Image Based on File Transfer Methods”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 1999.

[19] *W. H. Tam*, “A Human Face Tracker System Design Study Using the Technology of Digital Image Process”, master thesis, Department of Electrical Engineering, National Cheng-Kung University, 1999.

[20] *Jin-Hong Li*, “Image Based Digital Monitoring System”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 2000.

[21] *Jui-Hsiung Peng*, “Moving Frame Based Method to the Inspection of Assembled PCB”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 2000.

[22] *Ya-Wen Chisu*, “Motion Pursuit in Active Vision System”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 2001.

[23] *Kuo-Feng Chang*, “Image Processing for Multiple Moving Objects in Surveillance and Recognition”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 2002.

[24] *Gen-Wei Hu*, “Automatic Image Detection System for Human Shape Analysis”, master thesis, Department of Mechanical Engineering, National Chiao-Tung University, 2003.

[25] Steven F. Barrett; Cameron H. G. Wright; Harry Zwick; Michael Wilcox; Benjamin A. Rockwell; Espen Naess, 〝Efficiently tracking a moving object in two-dimensional image space〞, Journal of Electronic Imaging, Vol. 10, pp.785-793, July 2001

[26] *Chien Sung-I1* and *Sung Si-Hun*, 〝Adaptive window algorithm with four-direction sizing factors for robust correlation-based tracking〞, Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, pp.208-215, 1997

[27] *Rafael C. Gonzalez* and *Richard E. Woods*, 〝Digital Image Processing〞 (second Edition), Prentice Hall Inc., New Jersey, 2002

[28] *Choi Min-Seok ; Kim Whoi-Yul*, 〝A novel two stage template matching

method for rotation and illumination invariance", Pattern Recognition, volume: 35, Issue: 1, pp. 119-129, January, 2002

[29] *Fathy, M.; Siyal, M.Y.; Darkin, C.G.*; "A low-cost approach to real-time morphological edge detection" IEEE Region 10's Ninth Annual International Conference. Theme: 'Frontiers of Computer Technology'. Proceedings of 1994 , pp. 759 -762 vol.2, 22-26 Aug. 1994

[30] *Tommasini, T.; Fusiello, A.; Trucco, E.; Roberto, V.;* "Making good features track better", Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pp. 178 -183, 23-25 June 1998

[31] *Snidar, L.; Foresti, G.L.,* "Real-time thresholding with Euler numbers", Pattern Recognition Letters, Volume: 24, Issue : 9-10, pp. 1543-1554, June, 2003

[32] *Fathy, M ; Siyal, M.Y.,* "An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis", Pattern Recognition Letters Volume: 16, Issue: 12, pp. 1321-1330, December, 1995

[33] *Wu Quen-Zong; Jeng Bor-Shenn,* "Background subtraction based on logarithmic intensities", Pattern Recognition Letters, volume: 23, pp.1529-1536, 2002