Robotica (1998) volume 16, pp. 109–116. Printed in the United Kingdom © 1998 Cambridge University Press

Technical Report

Real-time object tracking from corners

Han Wang, Choon Seng Chua & Ching Tong Sim

School of Electrical and Electronic Engineering, Nanyang Technological University (Singapore) 639798
e-mail: hw@ntuix.ntu.ac.sg

(Received in Final Form: April 2, 1997)

SUMMARY

This paper reports a visual tracking system that can track moving objects in real-time with a modest workstation equipped with a pan-tilt device. The algorithm essentially has three parts: (1) feature detection, (2) tracking and (3) control of the robot head. Corners are viewpoint invariant, hence being utilised as the beacon for tracking. Tracking is performed in two stages of Kalman filtering and affine transformation. A technique of reducing greatly the computational time for the correlation is also described. The Kalman filter predicts intelligently the fovea window and reduced computation dramatically. The affine transformation deals with the unexpected events when there is partial occlusion.

KEYWORDS: Object tracking; Real time; Corner detection; Machine vision.

1. INTRODUCTION

The objective of tracking is concerned with pursuing an object of interest that moves randomly. with a video camera mounted on a pan-tilt device, the implemented algorithm will be able to detect motion of the moving object and then command the pan-tilt device to follow the object such that the object will always lie at the center of the camera.[1–7] The execution time poses a sever constraint in real time performance. Typical inter-frames processing delay can run up to milliseconds, and in order to achieve smooth tracking at real time, typically 25 Hz (called video-rate), various optimizations and improvements are needed. The success of a tracking activity depends on an efficient feature extraction algorithm. Further, the design of tracking strategies that will pursue desired objects closely also plays an important role.

Much recent work has concentrated on the development of control algorithms and architectures under the assumption that vision can provide the necessary position information to drive the system. Espiau *et al.* have used the task-function approach in order to tackle the problem of competing control tasks in visual servoing.[8,9] The control strategy is based on finding the image Jacobian of the visual task which relates the displacement of an image feature to camera displacements. Secondary tasks are incorporated by projecting their demands onto the null space of the Jacobian of the primary task, the visual task which relates the displacement of an image feature to camera displacements. The visual tasks explored to date have also explored this idea in the context of stereo visual servoing. Typical tracking schemes fall into the following categories: (1) Region based, (2) Contour based and (3) Point based fixation. When considering purely monocular cues, various general image features suggest themselves as likely candidates to provide the required position information for gaze control. Grey-scale correlation and variations of this theme are examples of the first category of tracking algorithm. Inoue *et al.*[10] developed a VLSI circuit to find minimum normalized absolute image differences in real time. This system, others based on its technology, and the work of Pahlavan *et al.*[11] discussed above, have demonstrated that correlation is an effective cue for smooth pursuit, and has the benefit of tracking can take place without a prior model of the targets appearance. Correlation, however suffers from two major disadvantages; (1) it is not invariant either to changes in view-point or to cyclorotation of the scene caused either by camera or object motion, and (2) it has little immunity to local occlusions. An example of the second category is the B-spline snake, developed by Kass *et al.*[12] They model the dynamics of such a contour under forces exerted by the attraction to image edges; as the image feature moves, the snake is drawn along with it. The advantages of occlusion insensitivity and view-point invariance are incorporated via templates which restrict the deformation of the contour to be an affine deformation of the planar template. These advantages, however require a lot of computation time which is almost impossible to achieve in real time without parallel processing or special hardware.

Finally the most obvious example of the third category is corner detection and tracking. Corner features are unreliable; they may provide corners for which short term matches can be made. Nevertheless, corner features are view-point invariant and also simple to extract from images. Algorithms for matching individual corners between frames requrie little or no knowledge at all about the overall motion of the object which gives rise to the corners, which means complex bootstrapping is not required.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

2. HARDWARE SETUP

The tracking system is comprised of a Sparc workstation, a pan-tilt device and a CCD camera with digitiser.

The pan-tilt system by Directed Perception fulfills the requirements well and a picture of the pan-tilt system is shown in Figure 1. Pan-Tilt Unit (PTU) by Directed Perception is a popular choice for low-cost, fast and accurate positioning of cameras and other payloads. It comes with a programmable controller that is both user-friendly and reliable. Besides it offers easy setup with standard serial communication.

The PTU provides precise control of axis speed and acceleration. Upper and lower speed limits define the boundary on non-stationary pan-tilt velocities. The base (start-up) speed specifies the velocity at which the pan-tilt axis can be started from a full stop without losing synchronization. To achieve axis speeds above base speed, acceleration changes are required. The pan-tilt controller uses trapezoidal acceleration and deceleration for speeds above the base rate and less than the maximum allowed speed. The pan-tilt controller provides on-the-fly position and speed changes. If the direction is changed on-the-fly, the controller manages all deceleration, direction reversal and acceleration to achieve the most recently specified target pan-tilt speed and acceleration rates.

3. IMAGE FEATURES EXTRACTION

Image features extraction involves techniques for extracting information from an image. It generally subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved. In simple static scenes like a black circular object in a white background, features extraction is a straightforward task using simple algorithm like thresholding. Processing time is not a constraint in this case. However, in real time visual processing that involves dynamic scenes, the task becomes complicated. This could account for features of interest being occluded or deformed into other shapes, or similar objects may exist and this requires object recognition, or variations in environment characteristics like light intensity and background.

Corner detection is considered to be point-based image features detection. Corner features are view-point invariant, relatively simple to extract from images. Algorithms for matching individual corner between frames need to know little or no information at all about the overall motion of the object which gives rise to the corners. However corner detectors are notoriously unreliable; they may provide corners for which short-term matches can be made, but will rarely detect any one corner so that it can be tracked over an extended period. Immunity to these local problems can be gained by considering clusters of corners over long sequences for which there can be a guarantee of at least some temporal continuity, but this raises the issue of how to describe collective position and motion of point clusters. A suggestion to overcome this problem is discussed later using affine structure.

The main difficulty of corner detection is to achieve simultaneously accuracy of localization, consistency of detection and low computational complexity, all desirable features for fixation in a real-time gaze control system. The corner detector developed by Wang and Brady[13] defines a corner where the smoothed image irradiance E in the fovea window satisfies the conjoint of:

$$\begin{cases} \Gamma = \left(\dfrac{\partial^2 F}{\partial \mathbf{t}^2}\right)^2 - S\,|\nabla F|^2 = maximum \\ \dfrac{\partial^2 F}{\partial \mathbf{n}^2} = 0 \\ |\nabla F|^2 > T_1, \quad \Gamma > T_2 \end{cases} \tag{1}$$

where S is a constant measure of image surface curvature varying with different differentiation masks, F is the intensity image after Gaussian smoothing, and T_1 and T_2 are user-defined thresholds. $\mathbf{t}$ and $\mathbf{n}$ are tangential and normal unit vector to the contour. Figure 2 shows part of a model train and the detected corners.

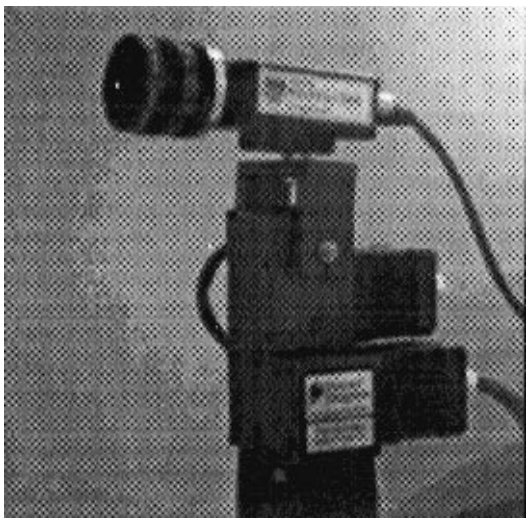

Fig. 1. The pan-tilt device.

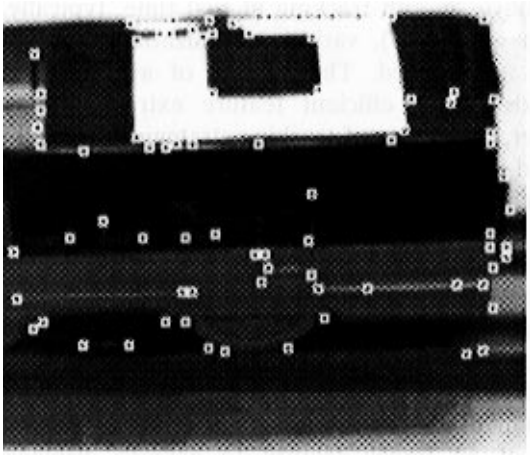

Fig. 2. Image of a model train with corners superimposed.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

4. IMAGE FEATURE MATCHING AND SPEEDUP METHODS

Template matching or so called correlation is a filtering method to detect a particular feature in an image. The template is, in effect, a sub-image that looks like the image of the object. Since the location of an object in an image is not known, template matching is performed at every pixel location. A similarity measure is then computed which reflects how well the image data matches the template for each possible template location. The point of best match is selected as the location of the feature.

The normalised cross correlation is adopted and it is less dependent on the local properties of the reference and input images than is the unnormalised correlation. The correlation coefficient $\gamma(i, j)$ is defined as

$$\gamma = \frac{\sum_{x,y}[f(x, y) - \bar{f}][w(x - i, y - j) - \bar{w}]}{(\sum_{x,y}[f(x, y) - \bar{f}]^2 \sum_{x,y}[w(x - i, y - j) - \bar{\omega}]^2)^{\frac{1}{2}}} \quad (2)$$

where f and w are the templates ($n = m \times m$, m is the template size) taken around the tracked feature in two consecutive frames; $\bar{f}$ and $\bar{w}$ are the local mean respectively. It is well known that the above equation is computationally expensive giving the task of real time matching. We propose two measures to reduce the computation without losing the quality of matching: (1) Introduction of theshold for matching quality; (2) sub-sampling

First, let's rearrange equation 2. Let D denote the denominator of equation 2,

$$D = \frac{1}{(\sum_{x,y}[f(x, y) - \bar{f}]^2 \sum_{x,y}[w(x - i, y - j) - \bar{w}]^2)^{\frac{1}{2}}},$$
$$t = n\bar{f}\bar{w}$$

we have

$$\gamma = \left(\sum fw - t\right)D \quad (3)$$

By definition, $\gamma \in [-1, 1]$. For $\gamma = 1$ there is a perfect match and with the increase of noise γ decreases. For $\gamma = 0$ and below, the two patterns are irrelevant. Notice the fact that $D > 0$, we can set a threshold such that when $\sum fw - t < 0$ or $\sum fw < t$, the matching process terminates, hence avoid computing the computationally heavy component D! We reduced the computation time by taking advantage of the fact that among the many potential matching candidates, only one is the best match.

The second measure of reducing the computation time for correlation is by means of sub-sampling. For example, a 7×7 template requires 49 pixels and the sub-sampled template would require only $4 \times 4 = 16$ pixels. However, this method has to be applied with caution, since over-do it would cause great lose of signal and resulting in unstable correlation. Our experiments have shown that the correlation time can be reduced greatly and the quality of matching is comparable with the traditional correlation method in equation 2. Incorporating the Kalman filter, we can further improve the timing by reducing greatly the number of potential candidates for matching.

Some remarks on correlation based matching

Although correlation provides a faster mean of performing objective recognition, it has the disadvantage of not invariant either to changes in view-point caused either by camera or object motion. In other words, obtaining correlation for changes in size and rotation can be difficult. Normalizing for size involves spatial scaling, a process that in itself adds a significant amount of computation. Normalizing for rotation is even more difficult. If a clue regarding rotation can be extracted from $f(x, y)$, then $w(x, y)$ is simply rotated so that it aligns itself with the degree of rotation in $f(x, y)$. However, if the nature of rotation is not known, looking for the best match requires exhaustive rotations of $w(x, y)$. This procedure is impractical and, as a consequence, correlation is seldom used in cases when arbitrary or unconstrained rotation is present. Normally dynamic correlation is performed on continuous image sequence rather than on a few random images. In real time object tracking, a sequence of frames is captured at video rate and correlation matching is performed on every frame of the sequence. The template for matching is predetermined from the first frame. To minimize the effect of rotation and changes in object size, the template is updated dynamically with the subsequent image that is matched correctly. With this method, the template will contain the latest information or sub-image of the moving object even though the particular object has gone through rotation or scaling.

5. KALMAN FILTERING AND PREDICTION

Cross correlation provides a low level confidence measure of the matching strength. However, the computation can be very time consuming since the template has to match blindly against every position obtained from the the corner extraction algorithm over the entire image. Kalman Filter,[14–17] can reduce this problem by predicting the possible location of the tracked points. This section describes a linear Kalman filter that monitors and tracks points in the image space. The system state space is defined as

$$\mathscr{X}(k) = [x(k)\dot{x}(k)y(y)\dot{y}(k)]^T \quad (4)$$

where x and y are the Cartesian coordinates of an object, $\dot{x}$ and $\dot{y}$ are the image velocity of the object at frame k. A random signal model is defined as

$$\mathscr{X}(k+1) = \underbrace{\begin{bmatrix} 1 & t_{k+1} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t_{k+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}(k)} \mathscr{X}(k) + \underbrace{\begin{bmatrix} 0 \\ u_1(k) \\ 0 \\ u_2 k \end{bmatrix}}_{\mathbf{w}(k)} \quad (5)$$

where the noise term $u_1(k)$ and $u_2(k)$ represent the change in pan and tilt velocity respectively over an interval t_{k+1}. Assuming images are taken in equal time

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

interval, then $t_{k+1} = T$ and $\mathbf{A}(k) = \mathbf{A}$. Let $E[\cdot]$ denote the mathematical expectation, then $E[\mathbf{w}(k)\mathbf{w}(k)^T] = \mathbf{Q}(k)$ represents the systems noise covariance matrix. As an object travels in all direction, the input noise $u_1(k)$ and $u_2(k)$ assume Gaussian distribution with zero mean and they are in general uncorrelated. The camera is assumed to provide noise estimates of the object location. The output of camera measurement is defined as

$$\mathcal{Y}(k) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}} \mathcal{X}(k) + \underbrace{\begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}}_{\mathbf{v}(k)} \tag{6}$$

where the additive noise, $\mathbf{v}(k)$ is assumed to be Gaussian with zero mean and $E[v(k)\ v(k)^T] = \mathbf{R}(k)$ gives error covariance of the noise sequence.

Let $\mathbf{P}(k)$ denote the state covariance matrix and $\mathbf{P}(k+1 \mid k)$ denote the prediction. The vector Kalman filter estimator is found as

$$\hat{\mathcal{X}}(k+1) = \mathbf{A}\hat{\mathcal{X}}(k) + \mathbf{K}(k+1)[\mathcal{Y}(k+1) - \mathbf{CA}\hat{\mathcal{X}}(k)] \tag{7}$$

where the filter gain $\mathbf{K}(k+1)$ is given by

$$\mathbf{K}(k+1) = \mathbf{P}(k+1 \mid k)\mathbf{C}^{\mathrm{T}}(\mathbf{CP}(k+1 \mid k)\mathbf{C}^{\mathrm{T}} + \boldsymbol{R}(k)]^{-1} \tag{8}$$

and the state covariance is updated by

$$\mathbf{P}(k+1) = \mathbf{P}(k+1 \mid k) - \mathbf{KCP}(k+1 \mid k) \tag{9}$$

Having defined the above equations, the Kalman filter equations are rearranged into the following 2 stages for computation:

(i) Prediction

$$\begin{aligned} \hat{\mathcal{X}}(k+1 \mid k) &= \mathbf{A}(k)\hat{\mathcal{X}}(k \mid k) \\ \mathbf{P}(k+1 \mid k) &= \mathbf{A}(k)\mathbf{P}(k)\mathbf{A}^{\mathrm{T}}(k) + \mathbf{Q}(k) \end{aligned} \tag{10}$$

(ii) Updating (where $k \Leftarrow k+1$)

$$\begin{aligned} \hat{\mathcal{X}}(k \mid k) &= \mathbf{A}\hat{\mathcal{X}}(k \mid k-1) + \mathbf{K}(k)[\mathcal{Y}(k) \\ &\quad - \mathbf{C}\hat{\mathcal{X}}(k \mid k-1)] \\ \mathbf{P}(k \mid k) &= \mathbf{P}(k \mid k-1) - \mathbf{K}(k)\mathbf{CP}k \mid k-1 \\ \mathbf{K}(k) &= \mathbf{P}(k \mid k-1)\mathbf{C}^{\mathrm{T}}(\mathbf{CP}(k \mid k-1)\mathbf{C}^{\mathrm{T}} + \mathbf{R}(k)]^{-1} \end{aligned} \tag{11}$$

5.1. Application of Kalman filter to object tracking

Consider a moving object has its positions measured by a camera. The initial prediction of the object location in $T = 1$ is arbitrarily chosen, based on the positional observation from the first two frames:

$$\hat{\mathcal{X}}(1) = [0, 1, 0, 1]^{\mathrm{T}}$$

Before finding the prediction of the object location in the next frame $\hat{\mathcal{X}}(2)$, the following system noise covariance is established:

$$\mathbf{Q}(k) = E[w(k)w(k)^{\mathrm{T}}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 \end{bmatrix}$$

where σ_1^2 and σ_2^2 are the variances whose values assume a Gaussian probability function bounded by $\mathrm{M} = 5$. Further, let

$$\mathbf{P}(0) = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The setting of $\mathbf{P}(0)$ is empirical. From equation (10), the values of $\hat{\mathcal{X}}(2 \mid 1)$ and $\mathbf{P}(2 \mid 1)$ are computed. In the updating stage, equation (11) is used to find $\mathbf{K}(k)$, followed by $\mathbf{P}(2 \mid 2)$ and from which the predicted object location in the next frame $\hat{\mathcal{X}}(2 \mid 2)$ is obtained. These values are then used to recursively generate the next set of data.

Kalman filter operation provides error analysis which is not available in Linear Predictor. The noise model used follows a Normal Probability Distribution function, where a desire point at boot time has confidence region of a few sigma (error covariance). Since a 2-D model is considered, the error covariance when plotted resembles an ellipse. As Kalman filter update process is in progress, it changes the error covariance matrix of the tracking point and if a point is constantly tracked, its error covariance will decrease. This means that the probability of that appears in the predicted location in the next frame is higher (lower error) and this gives rise to a smaller ellipsoid.

5.2. A comparison on linear prediction and the Kalman filter

A comparison on the performance between Kalman filter and Linear Predictor is shown in Figure 3. The simulation result is performed on an object moving in an ellipse path with added noise to the actual object location. Figure 4 gives the square of the prediction error between these two methods. It is clear that the Kalman filter predicts more accurately on the object location than

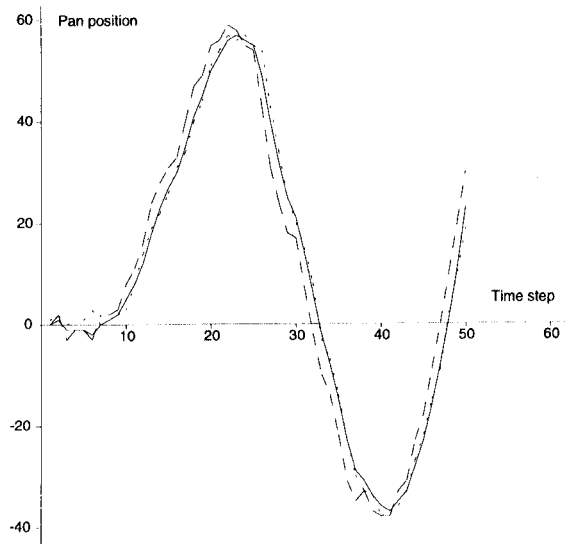

Fig. 3. Simulated results for the Kalman filter and Linear Predictor on a pan operation. Solid line-actual path: Dotted line—Kalman predictor; Dashed line—Linear predictor.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

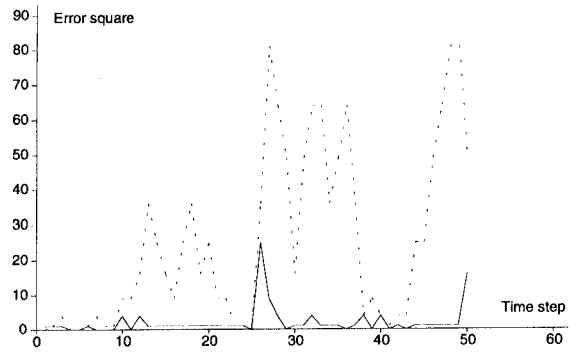

Fig. 4. Error square plot for the for Kalman filter and Linear Predictor on a pan operation. Solid line—Kalman filter; Dotted line—Linear predictor.

linear prediction. The simulation for the tilt motion is also carried out and shows the same behaviour.

6. AFFINE STRUCTURE

Tracking a moving object using center-of-mass method lacks consistency and often yields poor tracking performance especially when the object is occluded. Several prople have subscribed to the use of affine structure in image reconstruction, including reference 3 & 18. A temporal coherence between successive images forms the essence of affine structure. The affine structure method is simplified to take the advantage of monocular image tracking. For a coplanar image, it can be shown that using 3 basis points are sufficient to determine the fixation point accurately.[18] Besides, this method is shown to work well even though the object is sheared. This is true only if the object being tracked is a rigid body. This assumption is generally valid since most real world objects are rigid. In the case where the third basis point is not available, 2 basis points will serve the purpose although at a reduced accuracy.

6.1. Affine structure using 2 basis points

In case where 3 basis points information are not available, e.g. the tracked object is partially occluded, then we present a two basis points method to compute the fixation point. This method works under a further assumption, that is the projection is co-planar and the rotation is about the Z axis only. The fixation point determined using 2 basis points may not be as accurate when an object is sheared. In what follows, a weak camera projection can be justified for the fact that the distance from the camera is more significant compared to the object size. With this assumption, the error derived from finding the fixation point is negligible.

Referring to Figure 5, c is the fixation point of the initial object while c' is the required fixation point of the same object that had been rotated and translated (linear transformation). The motion can be defined as translation from o to o' plus a rotation of angle θ. The translation is given as

$$t = \tilde{o}' - \tilde{o}$$

and the rotation can be computed from the following

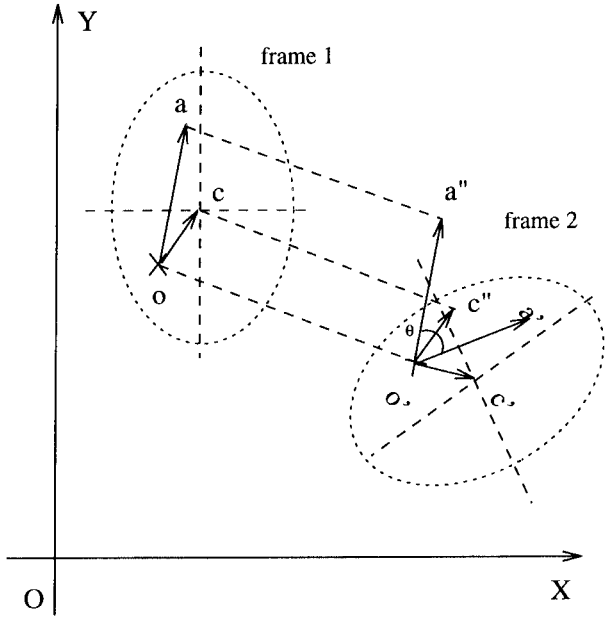

Fig. 5. Affine structure using the 2 basis points o and a.

condition

$$\theta = \angle a''o'a'$$

6.2. Transfer of the gaze direction

An important criteria for smooth tracking is that the same point on the target should be identified from frame to frame and used to generate the gaze demand. Under the affine stricture transfer approach, it happens to be relatively simple and straightforward even if the desired fixation point on the target is invisible or occluded.

An issue that needs to be tackled is when the object is occluded to an extent that only one point invisible. This problem is solved by assuming a weak perspective case. Then the fixation point is simply computed using translation operation and then in the subsequent frames, more basis points are hoped to be recovered when the object move out of the shade.

In the case of total occlusion, no information about the location of the object is available. Although using a bigger fovea window increases the chance of searching for object in the next frame, the delay due to visual processing is longer. More often than not, such delay results in the object moving out of that bigger fovea window if the speed of object is high. Instead of using a bigger fovea window, an attempt is made to predict the location of fovea window in the next frame by substituting the current measurement of object location with the predicted value. This attempt assumes that object travels at a constant velocity and such an attempt will fail when the object travels in opposite direction.

7. IMPLEMENTATION AND RESULTS

In this section, a real time object tracking example will be discussed in detail. This example will serve to illustrate how the system is integrated, together with various concepts and methodologies discussed in the earlier sections. In this example, the pan-tilt unit is

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

commanded to track a model train that is traveling on a track. This is a simplified version of real-time object tracking, *ie.* there is only one moving object in the fovea window.

Figure 6 shows a test scene of the model train traveling around the track. Two tracking algorithms are developed based on the moving corners but they differ in term of calculating the fixation point. The first algorithm uses the centroid of moving corner points as fixation point; whereas the second algorithm uses affine structure to calculate the fixation point.

7.1. Tracking using centroid of moving corners

The main feature of this algorithm is that the fixation point of the object is calculated from the centroid of the moving corners. In this implementation, a number of corners are selected as the active corners (typically seven). The fixation point of the object is the centroid of these active corners. Template for correlation matching (*ie.* 7×7 pixels in size) at each active corner is saved. Each active corner is assigned with a matching factor, *match* which indicates the number of time the particular active corner has matched with the image or not matched with the image. For example, a statement *match* = 3 means that the particular active corner has a matching corner for the past 3 frames. A negative match number will indicate that the active corner has failed to match for the past 3 frames. Normalized correlation matching is performed to determine how many moving corners have been successfully detected and matched. For each currently matched corner, the old template is updated with the newly matched corner template (by using the concept of dynamic correlation) and matching factor is increased by one. For the unmatched corner, the matching factor is decreased by one. A particular active corner will be dropped and replaced by another corner if it has not been matched for the past few frames (*ie.* 3). The centroid of all the matched corners will be the fixation point. The pan-tilt is commanded to move the pan and tilt axis to offset the position error (difference between the fixation point and center of the screen).

7.2. Tracking using affine structure

Tracking using affine structure gives rise to a more stable tracking performance compared with the tracking with centroid point. Implementation of the tracking algorithm for affine structure is similar to that for the tracking using centroid of the moving corners. The main consideration here is the selection of the basis point for the affine structure. The top three active corners with the highest match value will be assigned to affine structure. The purpose is to choose the older age of the track (*ie.* over how many frames this corner has been tracked, the likelihood being that older tracks are more relaible). If only two active corners are available, affine structure calculation for two points is used. In the case of only one active corner is qualified, translation on the fixation point based on that matched corner point is calculated. In the event of total occlusion or none of the active corners is matched, the predicted value for Kalman filter is used as fixation point. This process is repeated for a few subsequent frames (typically four) after which the target is considered lost.

7.3. Results

Both algorithms implemented are able to track the model train closely. A typical length of computation time for various routines within a frame is listed in the table. The computation time obtained is based on the model train tracking with seven active corners. A total time of 70 mSec is needed to process each frame, thus a rate of 15 Hz is achieved.

Instructions	Time (m Sec)
Image capture	10
Corner detection	35
Correlation etc.	25
Total	70

Trajectory plotted in Figure 7 and 8 compare the trajectory of the object relative to pan and tilt position for part of the train sequence. The trajectory plot shows that tracking using centroid of moving corners suffers from instability especially when the model train is moving around the curve (as shown at frame No. 125 for pan movement in Figure 7). This instability is caused by: a big jump of fixation point on the object from frame to frame, and an error in prediction fovea window due to incorrect fixation point used for Kalman filter prediction.

The main objective of tracking is to maintain the object trajectory at the center of the fovea window. Delay caused by motor stepping a visual processing results in the trajectory being deviated from the centre of image coordinate. A sinusoidal trajectory path is

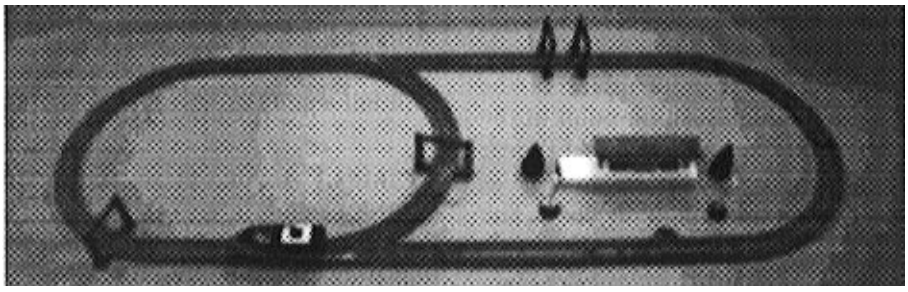

Fig. 6. A test scene of a model train traveling around a track.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

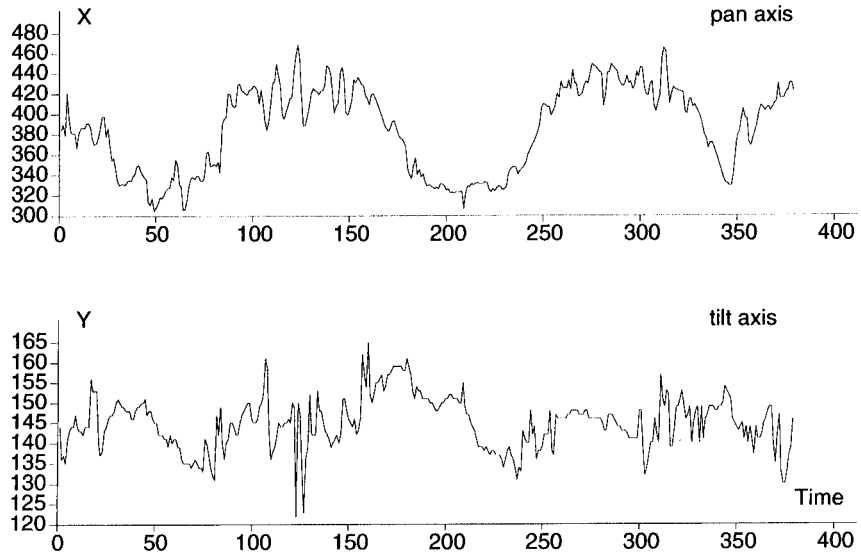

Fig. 7. Tracking with centroid.

obtained since the object is traveling around a elliptical track as shown in Figure 6.

Tracking using affine structure shows a relatively small deviation in the exact fixation location compared to the tracking algorithm using centroid of moving corners since the former provides a stable gaze direction. The possible errors in fixation point using affine structure are attributed to: the false match of active corners, and the shearing effects.

8. DISCUSSION

One of the aims in object tracking is to achieve smooth pursuit, possibly on any dynamic scene. Unfortunately, various assumptions due to hardware limitations and surrounding environments limit the types of objects being tracked. For example, visual processing delays can run up to milliseconds. As more corners are captured, longer processing time becomes intolerable to real-time requirements. Delays result in phase shift, lower bandwidth and subsequently instability. Therefore it is recommended that dedicated DSP hardware or parallel implementation is used to speed up the visual processing.

Falsely matches of a particular corner between two frames may generate a significance error in determining fixation point by affine structure. One possible method to reduce the probability of matching error is to restrict the corner selected in current frame for correlation matching. The new corner position in the new frame can be predicted using the Kalman filter prediction equations and a search neighborhood whose size is based on the predicted position covariance is defined. Closest corner detected in the neighborhood satisfying a correlation threshold is chosen to be best matched corner. However,

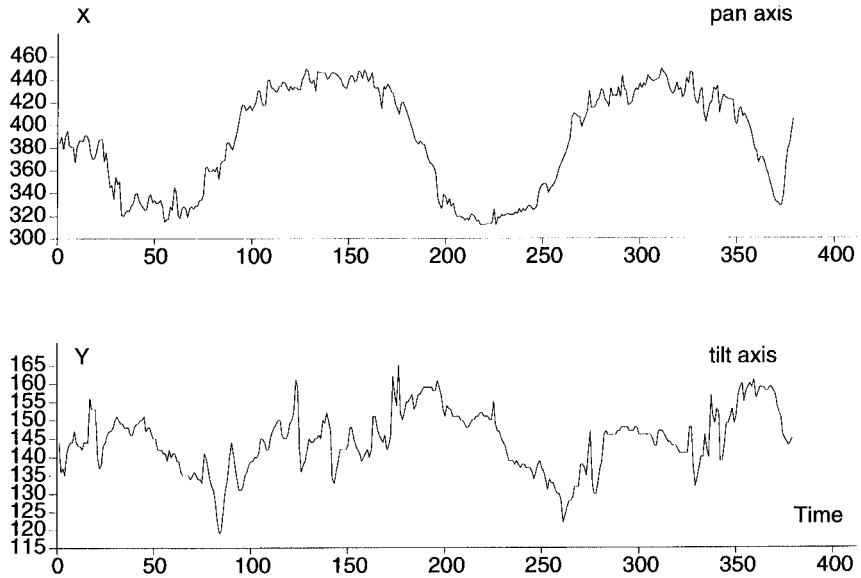

Fig. 8. Tracking with affine structure. Pan and tilt demands for two circular circuit of the train track. Tracking with centroid of moving corners suffer from instability especialy when the model train is moving around the curve as shown at frame No. 125 of the pan movement.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press

searching for every corner position using Kalman filter is time consuming and preferably to be done using a faster hardware.

The pan and tilt movements are accomplished using two stepper motors coupled in two orthogonal axes. It is noted that performance of the two motors differs due to the fact that the load is attached at the top of tilt axis. Thus, a higher base speed is used for tilt operation. Vibrations are also attributed to the characteristics of stepper motor which moves in steps. Besides if a high starting torque is achieved, then the amount of load it can carry would be larger. During real time operation, handshaking is seldom used due to time constraint. Such constraint means that velocity changes are seldom performed when the motor is working at high speed. In order to achieve smooth pursuit even at high velocity changes, a servo motor would fit this purpose well. Nevertheless, we found that the performance of our pan-tilt system is satisfying as no such high velocity changes are required.

9. CONCLUSION

In this paper, it has been shown that visual processing elements and active tracking algorithms are important for real-time object tracking. Satisfactory results in terms of speed, reliability and stability were achieved, even though the hardware used for visual processing was a modest Sun Sparc workstation. Hardware setup for the visual system and pan-tilt were successfully calibrated to give an optimum performance of the object tracking system. Kalman filter prediction was proven to give better accuracy in predicting the location of fovea window which contains the object compared to linear prediction. A real time features extraction method, corner detection was used to segment the moving object into a cluster of moving corners. These corners serve as interest points for correlation matching and provide a stable fixation point for gaze control system. Correlation matching was implemented successfully to match the moving corners from frame to frame. The success rate of matching is increased by introducing the concept of dynamic correlation. Affine structure concept was proved to be useful in providing a stable fixation point based on the available temporal continuity within a moving corner. The original concept of affine structure for 3 basis points was modified to cater for 2 basis points. This concept is important when the number of moving corners is reduced to two due to occlusion or unstable corners.

References

1. P.M. Sharkey, D.W. Murray, S. Vandevelde, I.D. Reid and P.F. McLauchlan, "A modular head/eye platform for real-time reactive action" *Mechatronics* **3**(4), 517–535 (1993).
2. A. Blake and A. Yuille, *Active Vision* (MIT) Press, Cambridge, Mass., 1992).
3. R. Cipolla, Y. Okamoto and Y. Kuno, "Robust structure from motion using motion parallax" *Proc. IEEE ICCV* (1993) pp. 374–382.
4. C.G. Harris, "Tracking with rigid models" **In:** *Active Vision* (A. Blake and A. Yuille, Eds.) (MIT Press, Cambridge, Mass., 1992) pp. 59–73
5. C. Brown, D. Coombs and J. Soong, "Real-time smooth pursuit tracking" *Active Vision* (A. Blake and A. Yuille, eds.) (MIT Press, Cambridge, Mass., 1992) pp. 123–136.
6. J.J. Clark and J.N. Ferrier, "Attentive visual servoing" **In:** *Active Vision* (A. Blake, and A. Yuille, eds.) (MIT Press, Cambridge, Mass., 1992).
7. K. Paklavan, T. Uhlin and J. Eklundh, "Integratting primary ocular processes" *Proc. ECCV'92 Second European Conference on Computer Vision,* Genova, Italy (May, 1992) pp. 526–541.
8. B. Espiau, F. Chaumette and P. Rives, "A new approach to visual servoing in robotics" *IEEE Trans. On Robotics and Automation* **8**(3), 313–326 (June, 1992).
9. C. Samson, B. Espiau and M. Le Borgne, *Robot Control: The Task Function Approach* (Oxford U.P., Oxford, U.K., 1991).
10. I. T. Tachikawa and M. Inaba, "Robot vision with a correlation chip for real-time tracking, optical flow and depth map generation" *Proc. IEEE Int. Conference on Robotics and Automation* (1992) pp. 1621–1626.
11. K. Pahlavan, T. Uhlin and J.O. Eklundh, "Dynamic fixation" *Proc. 4th Int. Conference on Computer Vision* (1993) pp. 412–419.
12. M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models" *Proc. 1st. Int. Conference on Computer Vision,* London (1987) pp. 259–268.
13. H. Wang and M. Brady, "A real-time corner detection algorithm for motion estimation" *Image and Vision Computing* **13**(9), pages 695–703 (1995).
14. S.M. Bozic, Digital and Kalman Filtering (Edward Arnold, London, UK, 1979).
15. Xiao-Rong Li and Yaakov Bar-Shalom, *Estimation and Tracking: principles, techniques and software* (Artech House, Inc., 1993).
16. S. Lee and Y. Kay, "A Kalman filter approach for accurate 3-D motion estimation from a sequence of computer images"*Computer Vision, Graphics and Image Processing: Image Understanding* **54**(2), 244–258 (Sept., 1991).
17. C.G. Harris and G. Stennett, "3-D object tracking at video rate-RAPiD *Proc. British Machine Vision Conference,* Oxford, UK (1990) pp. 73–78.
18. I.D. Reid and D.W. Murray, "Active tracking of foveated feature cluster using affine structure" *Int. J. Computer Vision* **18**(1), 41–60, 1996.

https://doi.org/10.1017/S0263574798000198 Published online by Cambridge University Press