The Active Recovery of 3D Motion Trajectories and Their Use in Prediction

Kevin J. Bradshaw, Ian D. Reid, *Member, IEEE*, and David W. Murray, *Member, IEEE*

Abstract—This paper describes the theory and real-time implementation using an active camera platform of a method of planar trajectory recovery, and of the use of those trajectories to facilitate prediction over delays in the visual feedback loop. Image-based position and velocity demands for tracking are generated by detecting and segmenting optical flow within a central region of the image, and a projective construct is used to map the camera platform's joint angles into a Euclidean coordinate system within a plane, typically the ground plane, in the scene. A set of extended Kalman filters with different dynamics is implemented to analyze the trajectories, and these compete to provide the best description of the motion within an interacting multiple model. Prediction from the optimum motion model is used within the visual feedback loop to overcome visual latency. It is demonstrated that prediction from the 3D planar description gives better tracking performance than prediction based on a filtered description of observer-based 2D motion trajectories.

Index Terms—Active vision, active camera platform, visual tracking, ground plane motion, interacting filters, prediction.

1 Introduction

Surveillance promises to be a major area of application for machine vision, where a wide variety of tasks ranging from the monitoring of traffic to the monitoring of secure areas may be fully or, more likely on commercial time scales, partially automated. The attributes that make these tasks tedious for the human—large amounts of redundant visual data punctuated by events which require immediate attention—suggest that this area is one which will benefit from an active approach, where the vision system has control over where and how it views the scene and is able both to react to, and to focus computational resources upon, unexpected and expected events in the scene.

In recent papers, we have described a mechatronic approach to the construction of an active vision system [1] and the use of image motion both to initiate saccadic, rapid gaze-shifting [2], [3] and subsequently to achieve and maintain smooth pursuit or gaze-holding [4], [5], [6]. As highlighted by Brown [7], [8] an important issue in active systems using closed-loop visual control is delay or latency in the feedback loop. In our multiprocessor vision system, even though the relatively intensive processes run at the video frame rate of 25Hz, the pipeline latency of between 100ms and 200ms is enough to cause the response to appear sluggish. A number of control schemes have been suggested to mitigate the problem [7], [9], [10]. One method we have explored is the explicit use of image motion and latencies measured frame by frame to predict "prompt" visual output, using either constant velocity or constant acceleration image motion filters [3], [10]. With natural targets, constant velocity filtering proved the more reliable.

An inherent difficulty with this image- or observer-based approach is that, because of the effects of viewpoint, object motion which can be described by a compact time-invariant description in three dimensions cannot be so described when projected into two. An obvious example is the viewing of 3D circular motion from an arbitrary viewpoint. This paper is therefore concerned first to recover 3D trajectories and then to use these for prediction in the visual feedback loop. Though of little direct concern, there is evidence that a similar interpretation of 2D motion in terms of 3D occurs in the human visuo-control system [11].

Using an active system with two cameras, the most obvious method of recovering 3D trajectories is to track stereoscopically and to utilize proprioceptive data from the camera platform to recover depth by triangulation. However, it turns out that achieving a stable and consistent position of attention in the scene itself presents significant challenges [12]. In this paper we take a different approach, recovering 3D trajectories monocularly by imposing the scene constraint that motion occurs in a plane—typically the ground plane. The ground plane constraint is frequently incorporated in surveillance techniques, such as in the model based recognition work of Tan et al. and Sullivan [13], [14], the motion description studies of Mohnhaupt and Neumann. [15] and the work of Nagel and coworkers [16], [17]. Intille and Bobick's [18] rectification of the motion of football players has some similarities with this work, though they track multiple targets off-line from broadcast footage using semantic knowledge.

The fact that motion models in 3D are more compact should make it easier to distinguish between a number of simple "canonical" motions, and to choose which best describes the current 3D motion. We have implemented Extended Kalman filters (EKFs) for each of a set of motion models, have allowed these to compete via the Interacting Multiple Model (IMM) first proposed by Blom [19], and

- *The authors are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK. E-mail: dwm@robots.ox.ac.uk.*

Manuscript received June 26, 1995; revised Nov. 18, 1996. Recommended for acceptance by A. Singh.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96122.

have then utilized the output from the model to predict prompt output in the visual feedback loop. In a real-time implementation, we demonstrate that prediction based on 3D does indeed give improved performance over 2D image-based prediction.

In the next section we outline briefly the active system and visual processing used to track using a single camera. In Section 3 we describe the calibration of projective plane-to-plane mapping which permits reconstruction of 3D motion trajectories in the world coordinate frame. Two methods are given, based on points and lines, respectively. Section 4 details the individual filters used to describe the trajectories, and Section 5 explains their combination in the IMM. Section 6 gives a comparative demonstration of performance using 3D prediction against that using image measurements alone.

2 Obtaining Observer-Based Trajectories

The principal elements in our active vision system—camera platform, vision system, gaze controller and servo controller—are shown in Fig. 1. The platform is a two camera device with four degrees of rotational freedom. For the work described in this paper however we utilize only one camera, using rotations about the elevation axis θ_e and one of the vergence axes θ_v. The latter is always orthogonal to the elevation axis, as will be clear from the forward kinematics required later. The vision system comprises parallel pipelines of transputers allowing several vision processes to offer results to the gaze controller concurrently.

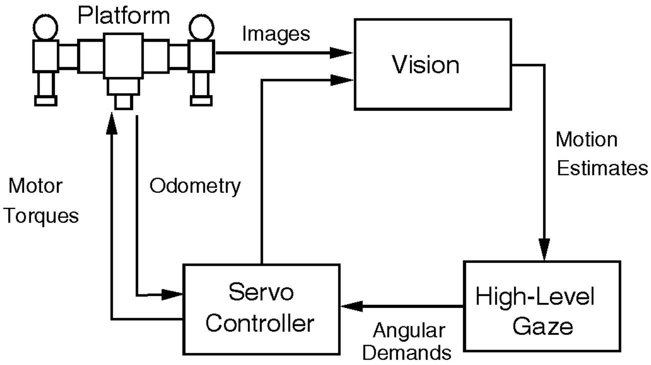


Fig. 1. The architecture of the visuo-control loop in our active camera system. Images from the platform cameras are processed by a number of parallel low-level vision algorithms. Vision results are sent to the high-level gaze controller which selects the visual feedback appropriate to the current task and generates a synchronous angular demand to the servo-controller.

Previous publications have demonstrated, inter alia, the generation of saccades where the gaze direction of the system is rapidly redirected to a target of interest, followed by periods of smooth pursuit where fixation is maintained upon a target [3], [6]. The gaze controller, implemented as a finite state machine, selects and reselects these behaviors to provide robust performance in natural dynamic scenes over extended periods.

One of the housekeeping operations of the servo-controller is to propagate odometry—in particular the axis positions or *gaze angles* θ_e and θ_v—around the system. For an extended pursuit sequence, the sequence of gaze angles $(\theta_e(t), \theta_v(t))$ for the platform's elevation axis and one vergence axis defines an observer-based trajectory for the target. Our first aim here is to recover an observer trajectory as the camera pursues an object.

Various algorithms have been implemented to drive pursuit behavior using active cameras, such as correlation [20], [21], deformable templates [22], feature-based affine transfer [4], [5] and segmented optical flow [6]. The last is used here, though any of the variety of methods might be employed to equal effect.

Optical flow is recovered in a small central, or foveal, area of the image. Because of its small size (~6° field of view in both x and y), it is assumed that only one moving object is included but, because background may be visible, a process of segmentation is required. The initial data computed at 25Hz are edge-normal components $\mathbf{v}$ of the motion field $\dot{\mathbf{r}}$ derived from the image irradiance $E(x, y, t)$ using the motion constraint equation [23].

$$E_t + \dot{\mathbf{r}} \cdot \nabla E = 0$$

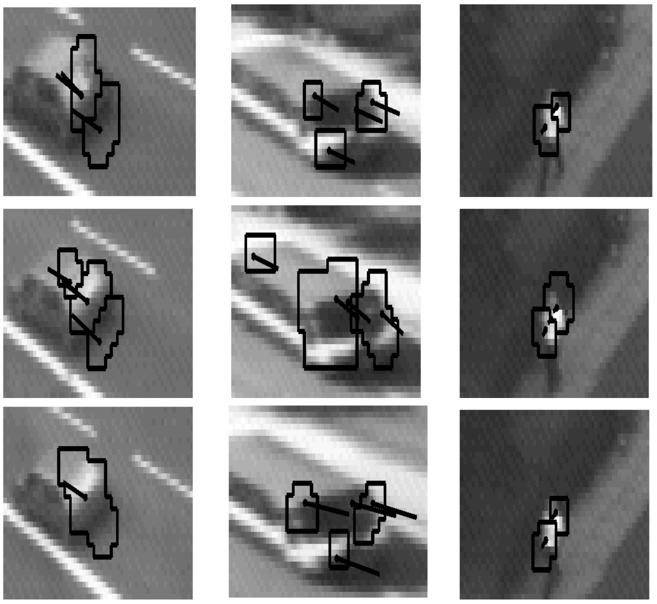
whence the components are found as

$$\mathbf{v} = -E_t \nabla E \,/\, |\nabla E|^2$$

Of course, whilst the camera pursues a target, a rotational image motion field $\dot{\mathbf{r}}_{rot}$ is induced throughout the image. Fortunately, such motion is independent of the scene depth, and so can be computed directly from the joint positions and velocities which are supplied by the servo-controller. It is subtracted from the observed motion to give components $\mathbf{v}_{ind}$ of the motion field $\dot{\mathbf{r}}_{ind} = \dot{\mathbf{r}} - \dot{\mathbf{r}}_{rot}$ arising from independently moving objects

$$\mathbf{v}_{ind} = \mathbf{v} - \left[\dot{\mathbf{r}}_{rot}\left(\theta_e, \theta_v, \dot{\theta}_e, \dot{\theta}_v\right) \cdot \hat{\mathbf{v}}\right]\hat{\mathbf{v}}$$

where $\hat{\mathbf{v}}$ is a unit vector in the direction of $\mathbf{v}$. Image regions which correspond to background will, within a noise tolerance, have $\mathbf{v}_{ind} \sim \mathbf{0}$, and are excluded from further consideration. Foreground motion regions are grown by spatially grouping the non-background vectors using a constant velocity model, and the motion vectors and their positions in a segmented region are combined to yield a mean position and mean velocity estimate, $\langle \mathbf{r}_{ind} \rangle$ and $\langle \dot{\mathbf{r}}_{ind} \rangle$. As is apparent in Fig. 2, multiple regions may be detected but, as noted earlier, we assume that they arise from the same object. (Other work on saccadic redirections of gaze direction [3] is based on a similar motion process running at coarse scale across the entire image and distinguishes between different regions using magnitude of motion, direction of motion, and area of the moving region. However, the "interest value" of a region based on these parameters remains hand-coded for different applications.) Our use of foveal motion for tracking is further described in [6].



(a) (b) (c)

Fig. 2. Optical flow detected, segmented and fitted to the foveal imagery while tracking (a) A car. (b) A bus. (c) A person. The number of independent motion regions detected at each frame can change, but that in this application all are assumed to be from the same object.

Our aim is to center the camera's viewpoint on the target, and to match its velocity. Because we have already subtracted the motion induced by camera rotation, $\langle \mathbf{r}_{ind} \rangle$ and $\langle \dot{\mathbf{r}}_{ind} \rangle$ are *demands* which could, after passing through the inverse kinematics, be sent to the joint servo-controller. However, these raw signals are delayed by the visual latency. As indicated in the introduction, one method we have used to reduce the effect of latency is filter the position and velocity using a constant velocity Kalman Filter, and use the product of filtered velocity and the known latency to evaluate a corrected, prompt, positional demand. Later in the paper we explore whether filtering using 3D motion is more effective.

Three examples of the output from the foveal optical flow process during tracking of a car, a bus, and a person are as shown in Fig. 2a through Fig. 2c. The camera platform was mounted on the sixth floor of an office block. In each case, the stills are every eighth frame from a 25Hz sequence.

The observer trajectories captured from several persons entering an leaving the University's Computing Center are shown in Fig. 3a, overlaid on an image taken from the rest frame. (Note that the rest-frame image is for graphical illustration only, and was neither captured during pursuit nor used in the analysis.) The actual observer trajectories from one of these pursuit episodes are shown in Fig. 3b and Fig. 3c. The person exits the building, proceeds along the pavement to their left for some 30m, turns around and doubles back before cutting across the grass to re-enter the building. The time axis unit is one frame. Each took 40ms, and so the pursuit sequence covers a period of some 56 seconds.

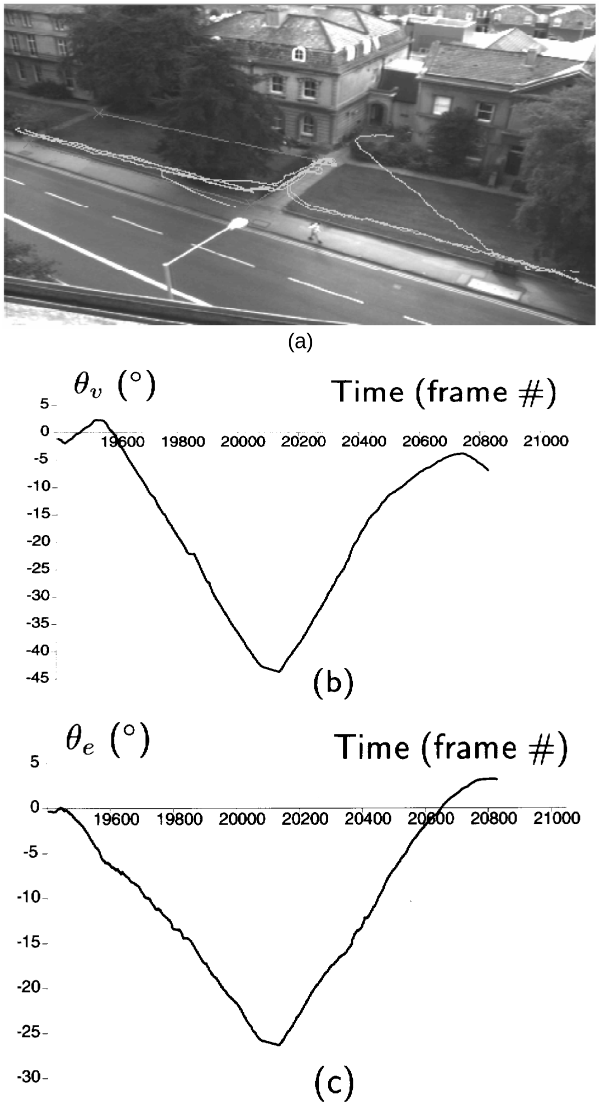


Fig. 3. Observer-based trajectories. (a) Overlaid on a rest-frame image. (b) and (c) Expressed for one trajectory as vergence and elevation joint angles θ_v, θ_e (degrees) versus time (in frame units of 40ms).

3 Projective Mapping to Ground Plane Coordinates

We now turn to the calibration method used to convert the observer trajectories $\theta_e(t)$, $\theta_v(t)$ of the sort shown in Fig. 3, into a trajectory describe in planar Euclidean coordinates based in a scene plane. The basic method requires no knowledge of the camera's position relative to the scene plane.

Fig. 4 shows that any pair of gaze angles can be mapped to a point $\mathbf{x}$ in the *frontal-plane*—a plane perpendicular to the resting gaze direction, $\theta_e = \theta_v = 0$, and an arbitrary distance in front of the rotation center of the camera. The point, given by the forward kinematics, is

$$\mathbf{x} = \left(\sec\theta_v \tan\theta_e, \qquad \tan\theta_v, \qquad 1\right)^{\mathrm{T}}$$

where we have chosen the arbitrary distance to be unity. The three-vector $\mathbf{x}$ is a homogeneous coordinate in the 2D frontal plane.

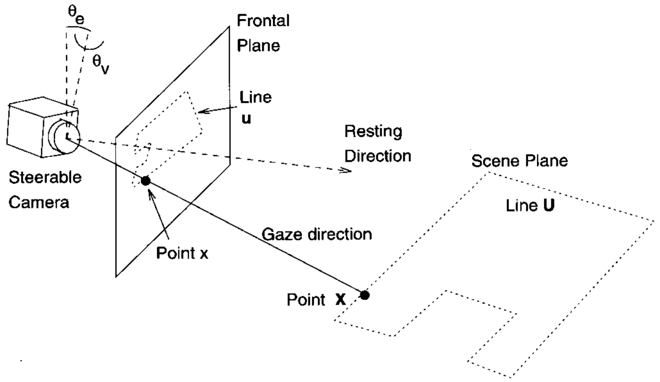


Fig. 4. Scene elements are projected into the frontal plane.

The point corresponding to **x** in the scene is **X**, where **X** is again a three-vector, a homogenous coordinate in a 2D scene plane. The pair of corresponding points on the two planes is related projectively by an homography

$$\mathbf{X} \equiv [\mathbf{M}]\mathbf{x}$$

where the 3×3 matrix [**M**] has only eight degrees of freedom because scale is arbitrary under projective equivalence. The homography [**M**] can thus be recovered by establishing the correspondence between at least four known points, or at least four known lines, in each of the two planes [24].

3.1 Recovery From a Four-Point Calibration

To establish the calibration using points, the active camera is directed to view in succession the calibration points whose scene coordinates $\mathbf{X}_i$ are known. The fronto-parallel plane position $\mathbf{x}_i$ is derived from the joints angles $(\theta_e, \theta_v)_i$. Each point correspondence provides an equation

$$\begin{pmatrix} \lambda_i X_i \\ \lambda_i Y_i \\ \lambda_i \end{pmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & 1 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

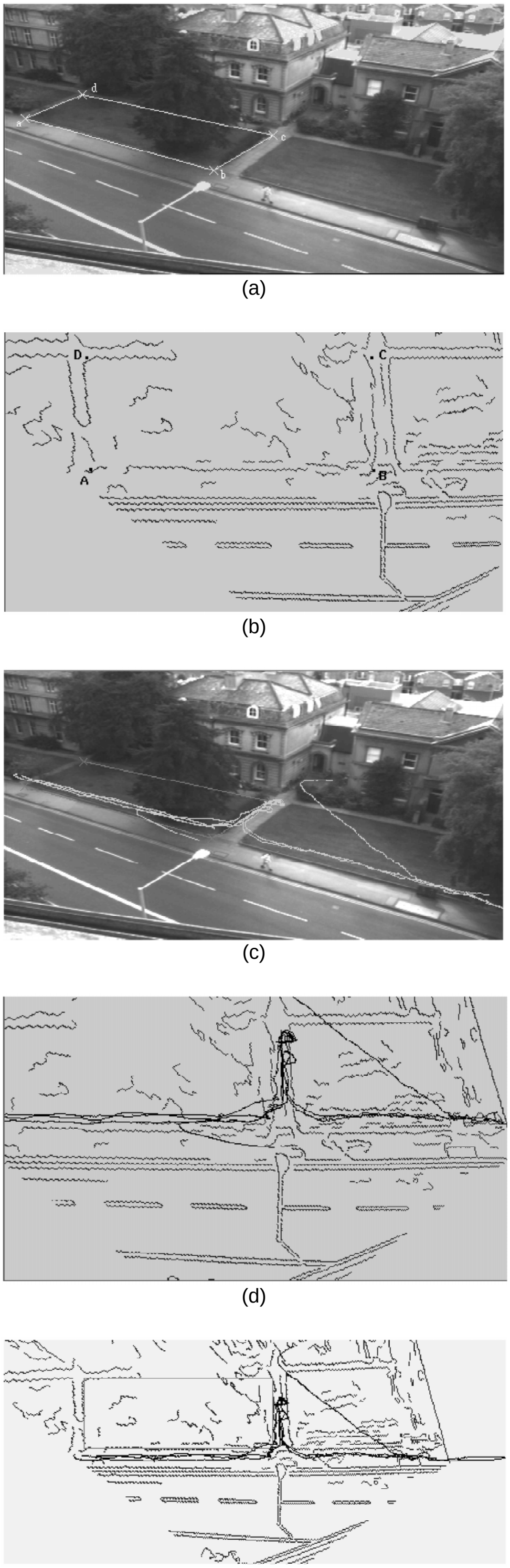
where by making the scale explicit we can enforce equality rather than projective equality, and where to constrain the degrees of freedom of [**M**] we set $M_{33} = 1$. Eliminating λ_i leaves two equations contributing to the system

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & 1 & 0 & 0 & 0 & -X_i x_i & -X_i y_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -Y_i x_i & -Y_i y_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} M_{11} \\ M_{12} \\ M_{13} \\ M_{21} \\ M_{22} \\ M_{23} \\ M_{31} \\ M_{32} \end{pmatrix} = \begin{pmatrix} \vdots \\ X_i \\ Y_i \\ \vdots \end{pmatrix}$$

or abbreviated $[\mathbf{A}]\mathbf{m} = \mathbf{b}$. For four or more points, the system can be solved in the least-squares sense as $\mathbf{m} = [\boldsymbol{A}^{\mathsf{T}} \boldsymbol{A}]^{-1} [\boldsymbol{A}^{\mathsf{T}}]\mathbf{b}$.

3.1.1 Implementation Example

The view obtained in the resting $\theta_e = \theta_v = 0$ direction from the window of the office block is given in Fig. 5a. Overlaid on the image are the four points used for calibration. In the scene these points $\mathbf{X}_i$ are the corners of a nominally rectan-



(a)

(b)

(c)

(d)

(e)

Fig. 5. Calibration of the outdoor scene. (a) Calibration points in rest-frame image. (b) Image features lines projected onto the ground plane. (c) Trajectories in the observer space. (d) Trajectories converted to the ground plane. (e) Corrected ground plane trajectories.

gular lawn measured as $(X, Y)_A = (0.0, 0.0)$, $(X, Y)_B = (21.1, 0.0)$, $(X, Y)_C = (21.1, 7.9)$, $(X, Y)_D = (0.0, 7.9)$ in the ground plane, where each unit is one meter. Centering the active camera on these points gave head joints angles of $(\theta_e, \theta_v)_A = (1.24°, 21.51°)$, $(\theta_e, \theta_v)_B = (–3.62°, 4.66°)$, $(\theta_e, \theta_v)_C = (–0.48°, –0.67°)$, $(\theta_e, \theta_v)_D = (3.53°, 16.40°)$, giving

$$[\mathbf{M}] = \begin{bmatrix} 0.228 & 0.601 & -73.281 \\ 0.135 & -0.479 & 51.757 \\ 0.002 & 0.013 & 1.000 \end{bmatrix}$$

By registering the image with the frontal plane it is possible approximately to overlay image features onto the calibrated plane. Fig. 5b shows this on edge features computed from the image in Fig. 5a. The recovered observer-based trajectories of people entering and exiting the building on the far side of the road and walking along the pavement are shown again for ease of comparison in Fig. 5c, and Fig. 5d shows the trajectories converted to ground plane trajectories using the computed homography.

If the pursued object lies above the ground plane, the 3D trajectory will be incorrect: it will be positioned where the gaze direction actually strikes the ground plane, that is, further from the camera. This problem is evident in the tracks of Fig. 5d—empirically it is found that it is the torso center that is tracked rather than the feet. (The same problem is evident in static features also: note that the outline of the street lamp, which is of course above the ground plane, is reconstructed as though it were an object "painted" on the ground plane.)

Strictly within the present method, the trajectory can be corrected only by recalibration from points at the correct height h, provided that h is constant. If no such points can be found, then the simplicity of the method is lost somewhat and the ground plane coordinates of the camera (X_c, Y_c) must be known, together with either the height Z_c of the camera, or a knowledge of the plane parallel to the ground plane and containing the camera's rotation center. Using the latter, if the current gaze direction makes an angle α with the parallel plane then the corrected ground position is

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} + \frac{h \cot \alpha}{\sqrt{\Delta X^2 + \Delta Y^2}} \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}$$

where $\Delta X = X_c - X$ and $\Delta Y = Y_c - Y$. Fig. 5e shows trajectories corrected using the latter method.

In Fig. 6 we show the recovered $X(t)$ and $Y(t)$ ground plane trajectories corresponding to the observer trajectories given in Fig. 3.

3.2 Recovery From a Four-Line Calibration

The calibration can also be achieved using four or more lines in the ground plane where, as the working below shows, there is no need to establish "endpoint-to-endpoint" correspondence.

The homography for lines is obtained from the dual relationships between lines and points in both planes, $\mathbf{u}^\mathrm{T}\mathbf{x} = 0$ and $\mathbf{U}^\mathrm{T}\mathbf{X} = 0$, as

$$\mathbf{U} = [\mathbf{M}^\mathrm{T}]^{-1}\mathbf{u}$$

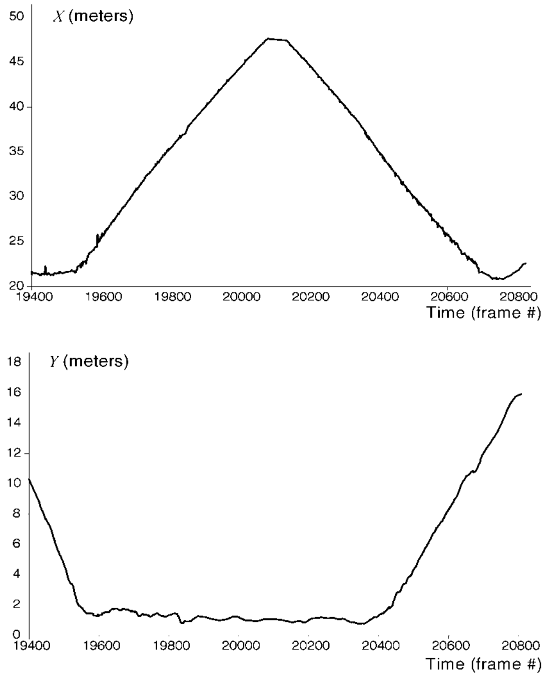


Fig. 6. The ground plane positions X and Y of the tracked person over some 56 seconds. Each frame lasts 40ms.

The homogeneous representation of a line in the ground plane is given in terms of the normal to the line $\hat{\mathbf{N}} = (\cos \Phi, \sin \Phi)^\mathrm{T}$ and the distance D (which may be negative) along the normal from the Cartesian origin to the line,

$$\mathbf{U} = (U, V, W)^\mathrm{T} = (\cos\Phi, \sin\Phi, -D)^\mathrm{T}$$

Similarly, for lines in the frontal plane,

$$\mathbf{u} = (u, v, w)^\mathrm{T} = (\cos\phi, \sin\phi, -d)^\mathrm{T}$$

To constrain the degrees of freedom we set $L_{33} = 1$. (The $L_{33} = 0$ condition occurs only if an observed line at the horizon of the ground plane is mapped to line through the origin of the frontal plane, an occurrence we can safely ignore.)

Each line correspondence is represented as

$$\begin{pmatrix} \lambda_i u_i \\ \lambda_i v_i \\ \lambda_i w_i \end{pmatrix} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{pmatrix} U_i \\ V_i \\ W_i \end{pmatrix}$$

and after eliminating λ_i contributes two equations to the system

$$[\mathbf{A}] \left(L_{11} L_{12} \ldots L_{31} L_{32}\right)^\mathrm{T} = \begin{pmatrix} \vdots \\ W_i u_i \\ W_i v_i \\ \vdots \end{pmatrix}$$

where

$$[\mathbf{A}] = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ U_i w_i & V_i w_i & W_i w_i & 0 & 0 & 0 & -U_i u_i & -V_i u_i \\ 0 & 0 & 0 & U_i w_i & V_i w_i & W_i w_i & -U_i v_i & -V_i v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

As in the earlier case, the unknown elements $\mathbf{l} = (L_{11}, \ldots, L_{32})^\mathrm{T}$ can be recovered by least-squares.

3.2.1 Implementation and Example Results

The implementation of the line calibration is somewhat more involved than that for points. The calibration lines are "traced" actively by the camera. The driving process is a video-rate implementation in the fovea of the Canny edge detection and hysteresis linking algorithms [25]. For each image frame, after computing edgels and linking them together into strings, the edgel nearest the center of the fovea is located and its parent string identified. The gaze controller then traverses the string in the current direction to find the *n*th neighbor of the central edgel, and sends its position as a demand to the servo-controller. Repeating this process as each image is received causes the camera to trace smoothly along an extended edge, even though no point correspondences are made between frames. Fig. 7 shows alternate frames from a 25Hz sequence of edgemaps obtained as the camera traces around a string.

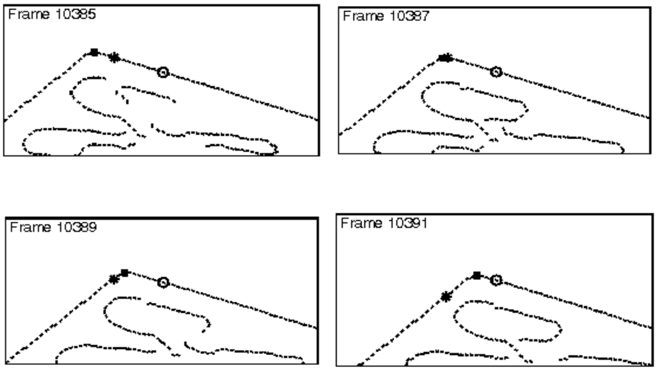


Fig. 7. Alternate frames from a 25Hz sequence of edgemaps obtained as the camera traces around a string. The near-center string is marked with a circle, the *n*th-neighbor edgel which is creating the positional demand is marked with a star. A point of high curvature on the traced string is marked with a square.

As line tracing proceeds, at each frame the gaze angles (θ_e, θ_v) obtained from the platform odometry are used to determine the intersection of the gaze direction with the frontal plane and the coordinates **x** are stored. When a point of high curvature is found, indicating the end of a line, the succession of stored **x** values are deemed to form a straight edge, a **u** is fitted. The fitted lines are stored and used for matching and calibration with the known world lines **U** to recover the homography.

Fig. 8a shows the observer trajectories obtained in the frontal plane while pursuing a radio-controlled buggy around the lab floor. The frontal image, which is included here for illustration and plays no part in the analysis, also shows part of the white-lined pattern used for calibration. The 50° field of view in the frontal image is much less than the near 180° field of view accessible to the active camera. The wider the field of view, the more accurate the calibration. Fig. 8b shows the Cartesian reconstruction of the trajectories and calibration pattern.

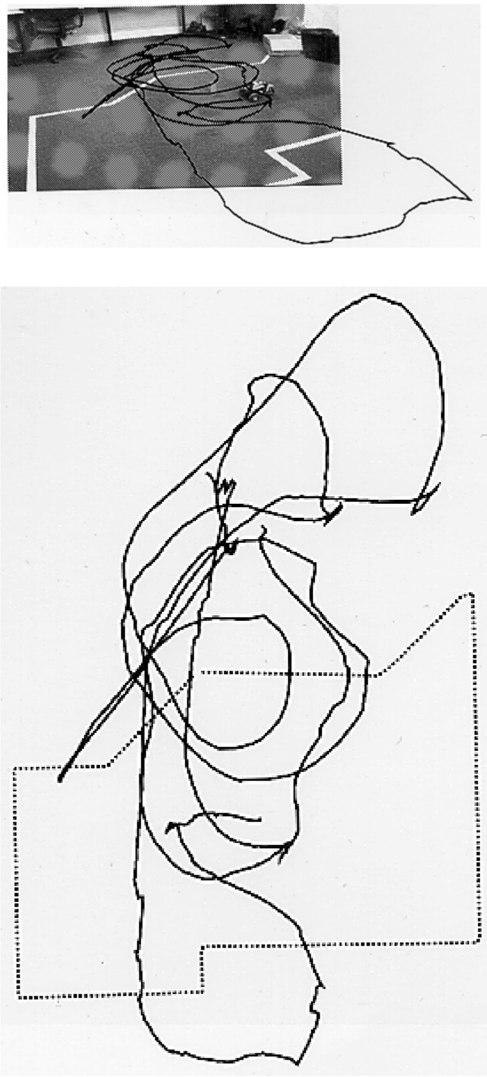


Fig. 8. (a) Buggy and observer track. (b) Ground plane trajectory.

4 Trajectory Filtering

We have developed filters embodying three simple canonical motions to describe the ground plane trajectories using the following scene models based on speed and direction, rather than individual speeds in the X and Y directions:

1. CSD Constant speed and direction, i.e., zero acceleration and turning rate.
2. CS Constant speed i.e., having possibly nonzero turning rate.
3. CD Constant direction i.e., having possibly nonzero acceleration.

Each model is incorporated in an Extended Kalman Filter (EKF), and the set embedded within the framework of the Interacting Multiple Model (IMM) [19] which selects the most appropriate filter to represent the target dynamics at any time.

4.1 The Individual Filters

For convenience, each of the filters for constant speed and direction (CSD), constant speed with turning rate (CS), and constant direction with acceleration (CD) has the state vector

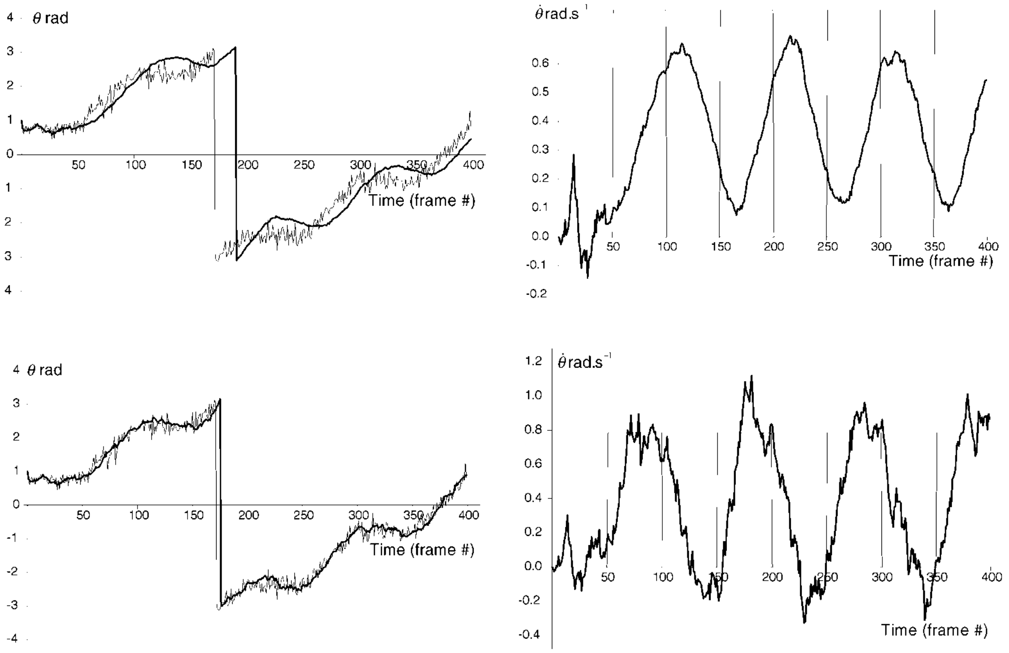


Fig. 9. The effects of increasing the process noise on estimated direction θ and turning rate $\dot{\theta}$ parameters for the CS filter. (a) Shows the temporal response when $\sigma^2_{\ddot{\theta}} = 1\text{rad}^2.s^{-4}$ and (b) Shows the response when $\sigma^2_{\ddot{\theta}} = 100\text{rad}^2.s^{-4}$. With higher process noise, the response is prompter, at the expense of increased noise in the turning rate estimates. The thinner lines represent data, the thicker the filter output. The discontinuity in direction at ±π is merely one of winding and does not affect the filter performance.

$$\mathbf{x} = \begin{bmatrix} X & Y & s & \theta & \dot{s} & \dot{\theta} \end{bmatrix}^{\mathrm{T}}$$

At each frame we obtain an estimate of the target position, but also allow for independent information about velocity to be included, giving an observation vector of

$$\mathbf{z} = \begin{bmatrix} X & Y & \dot{X} & \dot{Y} \end{bmatrix}^{\mathrm{T}}$$

The observations are related to the state vector by

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_k) + \mathbf{d}_k$$

where $\mathbf{h}$ is the vector-valued observation model and $\mathbf{d}_k$ is an uncorrelated, zero-mean, Gaussian noise sequence defined by $E[\mathbf{d}_k] = \mathbf{0}$ and $E\left[\mathbf{d}_i\mathbf{d}_j^{\mathrm{T}}\right] = \delta_{ij}\mathbf{R}$. The visual processes and robot dynamics implicitly and explicitly involved in the selection of a position in the image and its transfer onto the ground plane make it almost certain that these assumptions are violated. However, as the absolute size of noise in the Cartesian tracks, e.g., Fig. 5e, is small compared with variations we wish to account for using the different filters, the competition between filters should not be unduly biased.

In an EKF the prediction of state and variance are:

$$\hat{\mathbf{x}}_{(k+1|k)} = \mathbf{f}\left(\hat{\mathbf{x}}_{(k|k)}, \mathbf{u}_k\right) + \mathbf{e}_k$$

$$\mathbf{P}_{(k+1|k)} = \nabla\mathbf{f}\mathbf{P}_{(k|k)}\nabla\mathbf{f}^{\mathrm{T}} + \mathbf{Q}_k$$

where $\mathbf{f}$ is the vector of non-linear state transition functions, $\mathbf{u}_k$ is the control input which here is zero, and $\mathbf{e}_k$ is the process noise, an uncorrelated zero-mean Gaussian noise sequence with expectations $E[\mathbf{e}_k] = \mathbf{0}$ and $E\left[\mathbf{e}_i\mathbf{e}_j^{\mathrm{T}}\right] = \delta_{ij}\mathbf{Q}$. The update of state and variance after a new datum arrives are

$$\hat{\mathbf{x}}_{(k+1|k+1)} = \hat{\mathbf{x}}_{(k+1|k)} + \mathbf{W}_{(k+1)}\left[\mathbf{z}_{(k+1)} - \mathbf{h}\left(\hat{\mathbf{x}}_{(k+1|k)}\right)\right]$$

$$\mathbf{P}_{(k+1|k+1)} = \mathbf{P}_{(k+1|k)} - \mathbf{W}_{(k+1)}\mathbf{S}_{(k+1)}\mathbf{W}^{\mathrm{T}}_{(k+1)}$$

where the gain matrix and innovation covariance are, respectively

$$\mathbf{W} = \mathbf{P}_{(k+1|k)}\nabla\mathbf{h}^{\mathrm{T}}\mathbf{S}^{-1}_{(k+1)}$$

$$\mathbf{S}_{(k+1)} = \nabla\mathbf{h}\mathbf{P}_{(k+1|k)}\nabla\mathbf{h}^{\mathrm{T}} + \mathbf{R}_{(k+1)}$$

In the following derivations the time-step for information input is T and in each case there is no control input $\mathbf{u}_k$. In the CS and CD filters we introduce process noise to allow the system to adjust to changing parameter values. It is observed that larger values of process noise variance reduce the response time of the system to changes in the system state, at the expense of introducing increased noise into the estimated state values. The process noise values also account for errors introduced by the linearization of the EKF.

4.1.1 All Filters

First, for all the filters the vector of functions $\mathbf{h}$ is

$$\mathbf{h} = (X, Y, s\sin\theta, s\sin\theta)^{\mathrm{T}}$$

so that

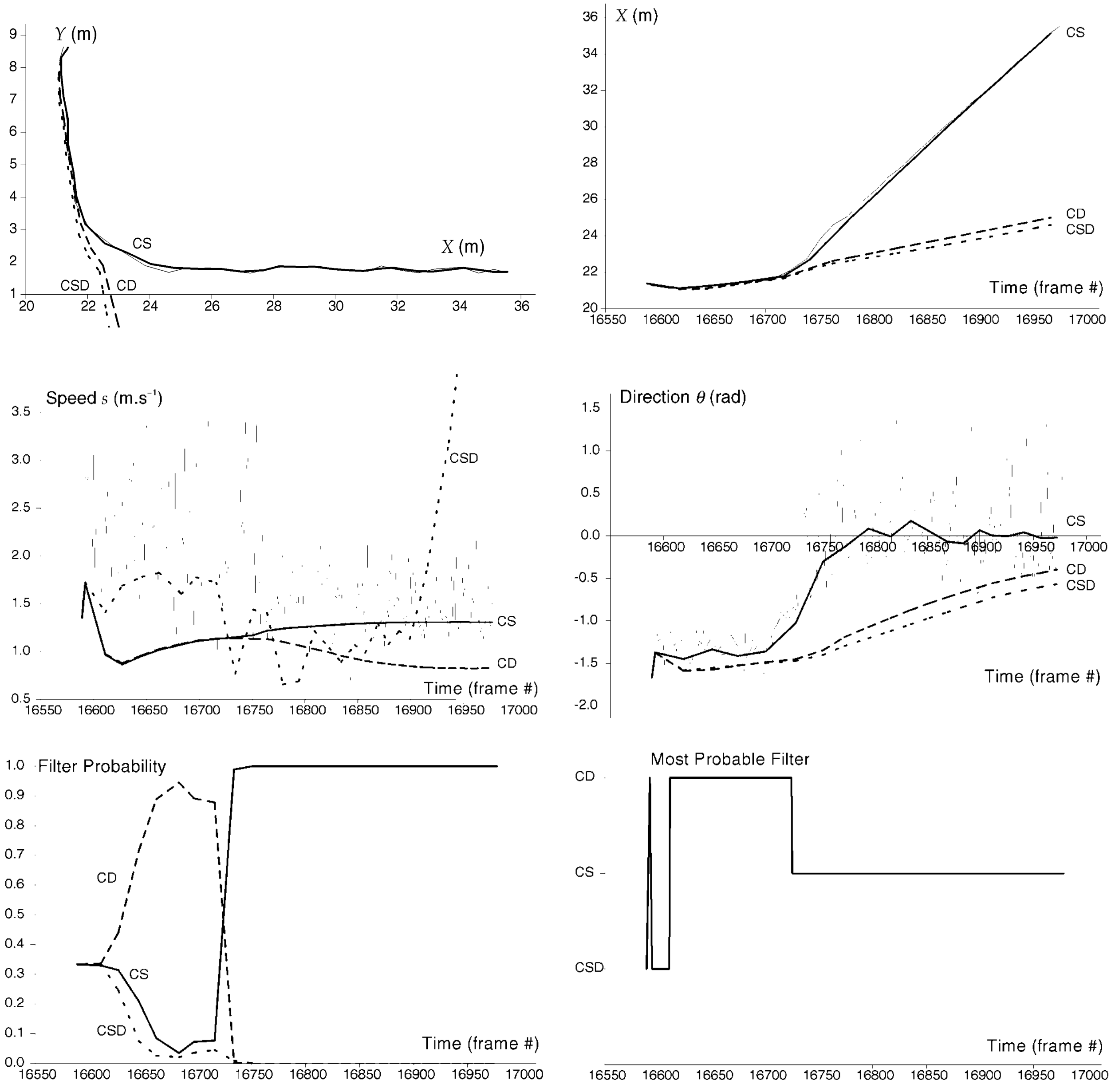


Fig. 10. Results for individual (i.e., non-interacting) EKFs for a walking person trajectory. (a) Shows the Cartesian trajectory (in meters) and (b)-(f) The filters' performance over time measured in frames of 40ms duration. In Fig. 10a-10e, the thin solid line records the measurements, and the thick solid, dashed, and dotted lines are outputs from the CS, CD, and CSD filters, respectively.

$$\nabla \mathrm{h} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin\theta & -s\sin\theta & 0 & 0 \\ 0 & 0 & \sin\theta & s\sin\theta & 0 & 0 \end{bmatrix}$$

4.1.2 The Constant Speed and Direction (CSD) Filter

Both s and θ are constant, so that the update equation over a time interval T is

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_k + s_k \sin\theta_k T \\ y_k + s_k \sin\theta_k T \\ s_k \\ \theta_k \\ \dot{s}_k \\ \dot{\theta}_k \end{bmatrix}$$

and the associated Jacobian matrix is

$$\nabla \mathbf{f} = \begin{bmatrix} 1 & 0 & \sin\theta T & -s\sin\theta T & 0 & 0 \\ 0 & 1 & \sin\theta T & s\sin\theta T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For this filter, $\dot{s}_k$ and $\dot{\theta}_k$ are not recovered and are included for computational uniformity in the IMM. Strictly, their values at any time step are noise terms drawn from zero mean Gauss-random sequences with variances $\sigma_{\dot{s}}^2$ and $\sigma_{\dot{\theta}}^2$:

$$\dot{s}_k = e_{\dot{s}} = N(0, \sigma_{\dot{s}})$$
$$\dot{\theta}_k = e_{\dot{\theta}} = N(0, \sigma_{\dot{\theta}})$$

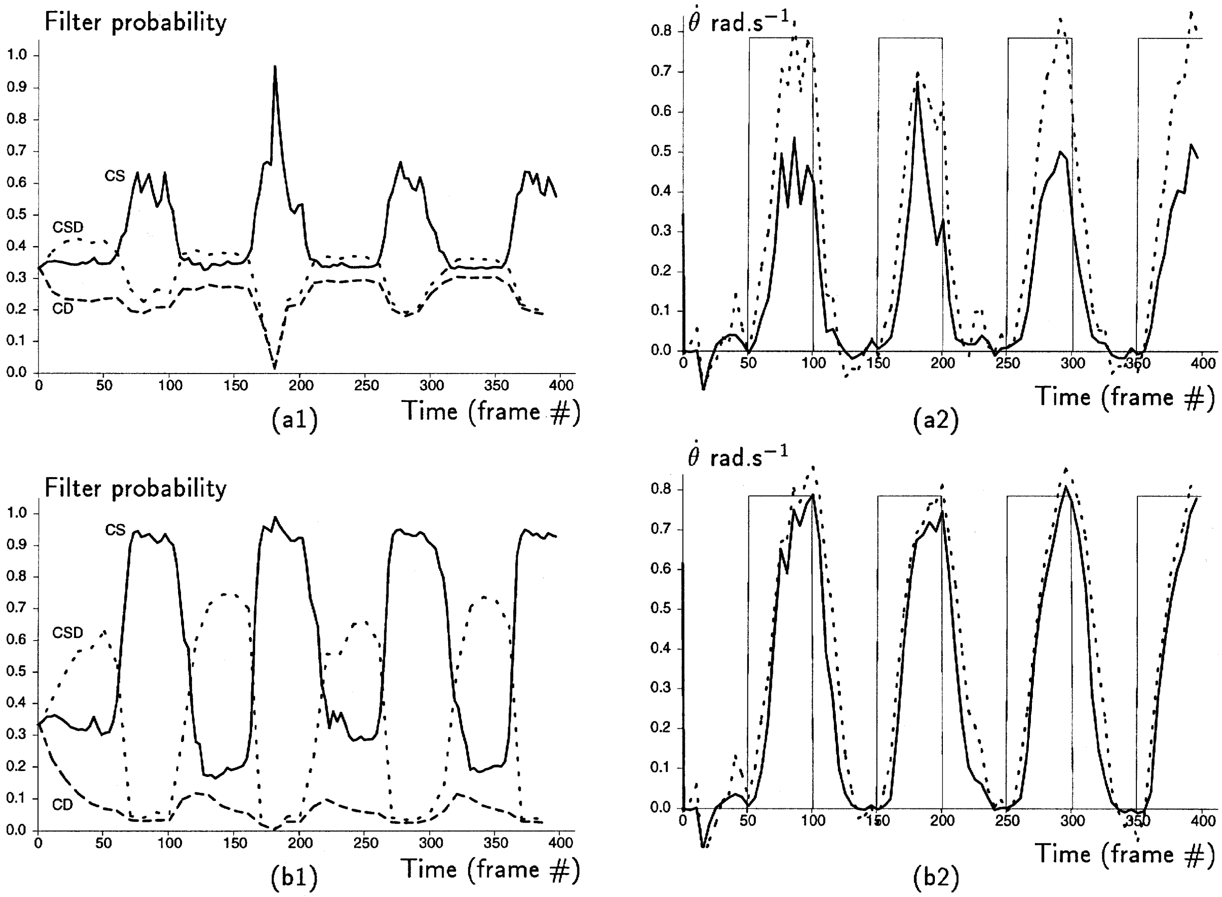


Fig. 11. The effect of the switching parameter on recovered turning rate and filter probability. (a) Shows the response when $J_d = 0.93$. (b) Shows that when $J_d = 0.99$. The discrimination between filter models is increased with a higher J_d, but response to change is lessened. The thick solid, dashed, and dotted lines are outputs from the CS, CD and CSD filters, respectively. The thin line in the turning rate diagrams is the veridical value.

To evaluate the process noise covariance, let both $e_{\dot s}$ and $e_{\dot\theta}$ be constant over a frame interval T, so that to first order the noise unmodeled by the update equation is

$$\mathbf{e}_k = \begin{pmatrix} \left(e_{\dot s}\sin\theta_k - e_{\dot\theta} s_k \sin\theta_k\right)T^2/2 \\ \left(e_{\dot s}\sin\theta_k - e_{\dot\theta} s_k \sin\theta_k\right)T^2/2 \\ e_{\dot s}T \\ e_{\dot\theta}T \\ e_{\dot s} \\ e_{\dot\theta} \end{pmatrix}$$

The covariance $\mathbf{Q}_k = E\left[\mathbf{e}_k\mathbf{e}_k^{\mathrm{T}}\right]$ is evaluated using the expectation values $E\left[e_{\dot s}^2\right] = \sigma_{\dot s}^2$, $E\left[e_{\dot\theta}^2\right] = \sigma_{\dot\theta}^2$, and $E\left[e_{\dot s}e_{\dot\theta}\right] = 0$. Again we stress that for the CSD filter only the first four members of $\mathbf{e}$ and the upper-left 4 × 4 sub-matrix of $\mathbf{Q}$ are strictly required for the EKF.

4.1.3 The Constant Speed (CS) Filter

The update equation is

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_k + \left(s_k/\dot\theta_k\right)\left[\sin\left(\theta_k + \dot\theta_k T\right) - \sin\theta_k\right] \\ y_k + \left(s_k/\dot\theta_k\right)\left[\sin\theta_k - \sin\left(\theta_k + \dot\theta_k T\right)\right] \\ s_k \\ \theta_k + \dot\theta_k T \\ \dot s_k \\ \dot\theta_k \end{bmatrix}$$

from which the Jacobian is found by differentiation. Here, $\dot s$ is not recovered but $\dot\theta_k$ is, so that we consider as unmodeled noise $e_{\dot s}$ and $e_{\ddot\theta}$, with variances $\sigma_{\dot s}^2$ and $\sigma_{\ddot\theta}^2$. In a similar manner to the CSD filter, we set these values to be constant over an update cycle to approximate the process noise vector $\mathbf{e}$ and hence the covariance matrix.

4.1.4 The Constant Direction (CD) Filter

Now θ is modeled as constant, but s may change, so that the update equation over a period T is

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} x_k + \sin\theta_k\left[s_k T + \dot s_k T^2/2\right] \\ y_k + \sin\theta_k\left[s_k T + \dot s_k T^2/2\right] \\ s_k + \dot s_k T \\ \theta_k \\ \dot s_k \\ \dot\theta_k \end{bmatrix}$$

Again the Jacobian is found by differentiation and the process noise covariance matrix is found by considering noises $e_{\ddot s}$ and $e_{\dot\theta}$ with variances $\sigma_{\ddot s}^2$ and $\sigma_{\dot\theta}^2$.

4.2 Tests of Individual Filters

The individual filters were first tuned by applying them to simulated noisy data generated from models each was designed to filter. An example of tuning the process noise in the CS filter is shown in Fig. 9. The synthesized motion has peri-

TABLE 1
ACTUAL AND ESTIMATED STATE DURATION, STATE ESTIMATES,
AND MEAN MODEL LIKELIHOOD FOR THE "ROUNDED SQUARE" TRAJECTORY

Most Likely Filter	Duration in Observations		Averaged Filter Likelihood			Averaged	
	Actual	Estimated	$\bar{p}_{CSD}$	$\bar{p}_{CS}$	$\bar{p}_{CD}$	$\dot{s}$	$\dot{\theta}$
						(m s^{-2})	(rad.s^{-1})
CSD	50	61	0.446	0.350	0.204	–0.006	0.025
CS	50	54	0.230	0.632	0.139	–0.002	0.616
CSD	50	46	0.441	0.360	0.199	–0.010	0.032
CS	50	57	0.201	0.667	0.132	–0.014	0.476
CSD	50	42	0.417	0.376	0.207	–0.008	0.046
CS	50	56	0.204	0.648	0.148	0.002	0.548
CSD	50	43	0.411	0.354	0.235	–0.008	0.023
CS	50	41	0.204	0.648	0.148	0.002	0.548

TABLE 2
ESTIMATED STATE DURATION, AVERAGED FILTER LIKELIHOOD
AND STATE ESTIMATES FOR THE WALKING PERSON

Most Likely Filter	Estimator Duration (Frames)	Averaged Likelihood			Average Values	
		$\bar{p}_{CSD}$	$\bar{p}_{CS}$	$\bar{p}_{CD}$	$\dot{s}$	$\dot{\theta}$
					(m s^{-2})	(rad.s^{-1})
CSD	11	0.358	0.310	0.333	0.002	-0.001
CD	2	0.367	0.248	0.385	1.196	0.001
CSD	1	0.400	0.243	0.357	0.006	0.001
CD	1	0.383	0.229	0.388	0.957	0.001
CSD	77	0.636	0.182	0.182	0.000	0.000
CS	72	0.259	0.605	0.136	0.000	0.441
CSD	170	0.547	0.293	0.159	0.000	0.000

ods consisting of 50 observations of constant speed, zero turn rate motion (i.e., straight line, no acceleration) followed by periods, again of 50 observations, of constant speed, fixed turn rate (45°s^{-1}). The target therefore moves in a square with rounded corners. Fig. 9a, shows that with a process noise value of $\sigma^2_{\ddot{\theta}} = 1 \text{ rad}^2.\text{s}^{-4}$, there is considerable overshoot in the estimated target direction but the recovered turning rate estimates are smooth. Increasing the process noise to $\sigma^2_{\ddot{\theta}} = 100 \text{ rad}^2.\text{s}^{-4}$ provides much more rapid response to maneuvers at the expense of increased noise in the recovered turning rate estimates. The chosen value of $30 \text{ rad}^2.\text{s}^{-4}$ was a compromise between promptness and smoothness.

Fig. 10 shows the results of applying each filter *individually* to part of the ground plane trajectory data from a walking person recovered by active tracking. The person exited the Computing Center, walked straight along the path, and then made a sharp left turn to walk along the pavement. We used measurement variance values of $\sigma^2_X = \sigma^2_Y = 0.1\text{m}^2$, $\sigma^2_{\dot{X}} = \sigma^2_{\dot{Y}} = 1.5\text{m}^2\text{s}^{-2}$. The initial covariances were taken as ten times the measurement noise values. Again we observe poor performance from the CSD and CD filters as the target maneuvers, but the CS filter performs well as the speed remains constant throughout the turn. These results were obtained with tuned values for process noises of $\sigma^2_{\ddot{s}} = 0.1\text{m}^2.s^{-4}$ and $\sigma^2_{\ddot{\theta}} = 1 \text{ rad}^2.s^{-4}$.

These examples serve merely to highlight the obvious dilemma faced in the choice of one filter. Using a strong filter, one has a firm basis for prediction but one loses responsivity to maneuvers. A weaker filter will work well all the time, but provides necessarily an uncertain basis for prediction.

This dilemma arises when observing any system which has dynamics which change abruptly and unexpectedly. Not surprisingly then, multiple-model-filtering has received considerable attention in work on radar tracking of aircraft. We have explored the Interacting Multiple Model (IMM) scheme for filter management, which from the literature appears to provide good state estimation in the presence of switching target dynamics with, importantly, a computational cost compatible with real-time operation [19], [26], [27], [28], [29], [30], [31].

5 THE INTERACTING MULTIPLE MODEL

The IMM algorithm operates with a set of N Kalman filters, each with a differing implicit motion model for the system dynamics. At each observation time-step, the outputs of the filters are first mixed according to a switching matrix **J** generated randomly from a distribution. The various filters are updated using the normal Kalman predict-update step. The likelihood of each filter is then evaluated, and sum of the state estimates of the filters, weighted by the respective filter likelihoods, formed as the output state of the system.

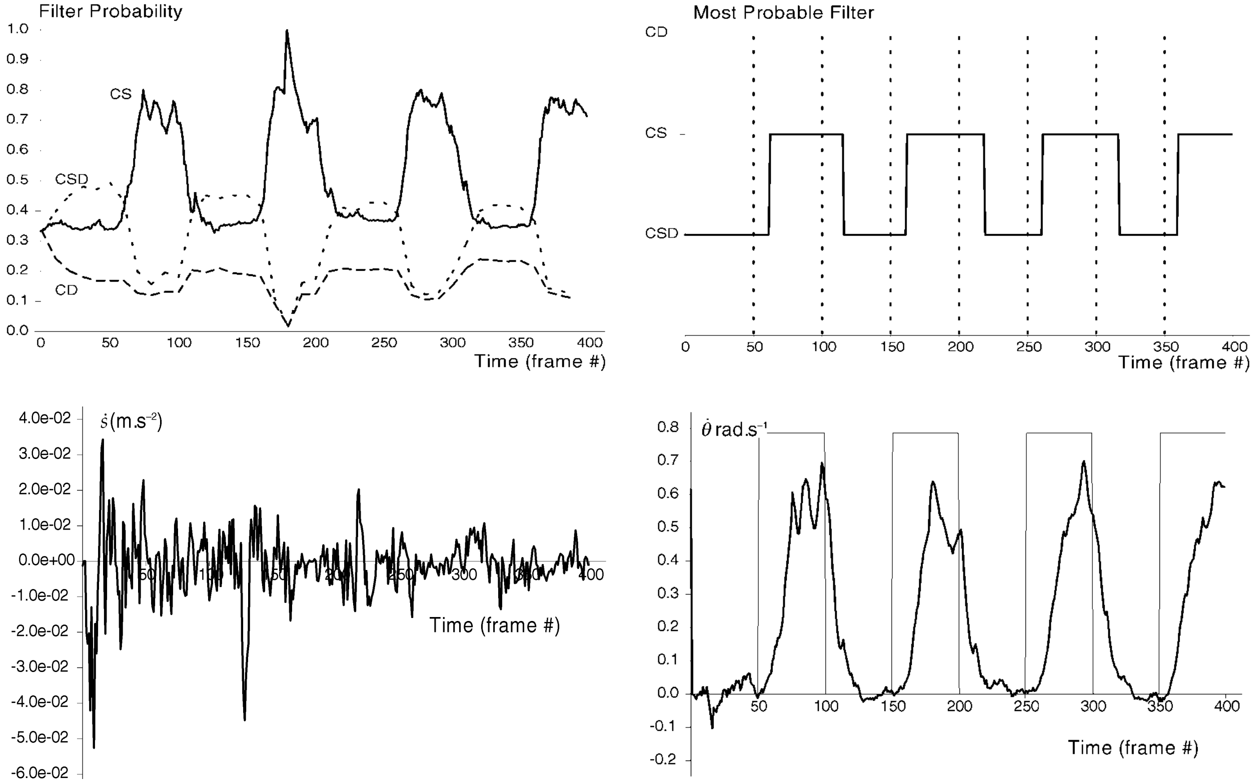


Fig. 12. IMM output for "rounded-square" trajectory data: (a) The filter likelihoods (where solid, dashed, and dotted lines are outputs from the CS, CD and CSD filters). (b) the most likely filter, where the dashed lines show when the actual maneuvers took place. (c) and (d). The overall IMM estimates of acceleration $\dot{s}$ in ms^{-2} and turn rate $\dot{\theta}$ in rad.s^{-1}. The thin line in (d) indicates the true value of $\dot{\theta}$.

At the end of time step k, let the ith filter, $i = 1, \ldots, N$, have state estimate $\hat{\mathbf{x}}_k(i)$, covariance $\mathbf{P}_k(i)$ and likelihood $p_k(i)$. The four steps in the IMM algorithm are:

1) For each filter model i, update the filter likelihoods to those appropriate to the start of the next step

$$p^*_{k+1}(i) = \sum_j J_k(ij)p_k(j)$$

Then mix the estimates of the state and covariance of each filter

$$\hat{\mathbf{x}}^*_k(i) = \frac{1}{p^*_{k+1}(i)}\sum_j J_k(ij)p_k(j)\hat{\mathbf{x}}_k(j)$$

$$\mathbf{P}^*_k(i) = \frac{1}{p^*_{k+1}(i)}\sum_j J_k(ij)p_k(j)\left(\mathbf{P}_j(j) + \left[\hat{\mathbf{x}}_k(j) - \hat{\mathbf{x}}_k(i)\right]\left[\hat{\mathbf{x}}_k(j) - \hat{\mathbf{x}}_k(i)\right]^{\mathrm{T}}\right)$$

2) The values of $\hat{\mathbf{x}}^*_k(i)$ and $\mathbf{P}^*_k(i)$ are used in the usual way in the EKF cycle; the prediction phase gives $\hat{\mathbf{x}}_{(k+1|k)}(i)$ and $\mathbf{P}_{(k+1|k)}(i)$ and the update phase gives $\hat{\mathbf{x}}_{k+1}(i)$ and $\mathbf{P}_{k+1}i$.

3) The likelihood of being in a filter state are updated as

$$p_{k+1}(i) = \kappa\frac{p^*_{k+1}(i)\exp\left[-\frac{1}{2}\mathbf{v}^{\mathrm{T}}_{k+1}(i)\mathbf{S}^{-1}_{k+1}(i)\mathbf{v}_{k+1}(i)\right]}{\left\|\mathbf{S}_{k+1}(i)\right\|^{1/2}}$$

where κ provides normalization $\sum_i p_{k+1}(i) = 1$. In this expression, the innovation vector is

$$\mathbf{v}_{k+1}(i) = \mathbf{z}_{k+1} - \mathbf{h}\left(\hat{\mathbf{x}}_{(k+1|k)}(i)\right)$$

and, as earlier,

$$\mathbf{S}_{k+1}(i) = \nabla\mathbf{h}\mathbf{P}_{k+1|k}(i)\nabla\mathbf{h}^{\mathrm{T}} + \mathbf{R}_{k+1}.$$

4) The usable output state and covariance $\hat{\mathbf{x}}_{k+1}$ and $\mathbf{P}_{k+1}$ are generated by taking a linear sum of the updated state and covariance estimates for each filter weighted by the updated filter likelihoods:

$$\hat{\mathbf{x}}_{k+1} = \sum_i p_{k+1}(i)\hat{\mathbf{x}}_{k+1}(i)$$

$$\mathbf{P}_{k+1} = \sum_i p_{k+1}(i)\left(\mathbf{P}_{k+1}(i) + \left[\hat{\mathbf{x}}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}\right]\left[\hat{\mathbf{x}}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}\right]^{\mathrm{T}}\right)$$

5.1 Multiple Model Tests

We now illustrate the benefits of the IMM approach, showing improved long-term filter response to motion trajectories, with the IMM detecting changes in the target motion state and selecting the appropriate filter accordingly.

The input mixing stage of the IMM performs filter mixing by resetting the state vector of each filter at each time-step to a linear combination of the output state vectors at the previous time-step, weighted by the filter likelihoods at

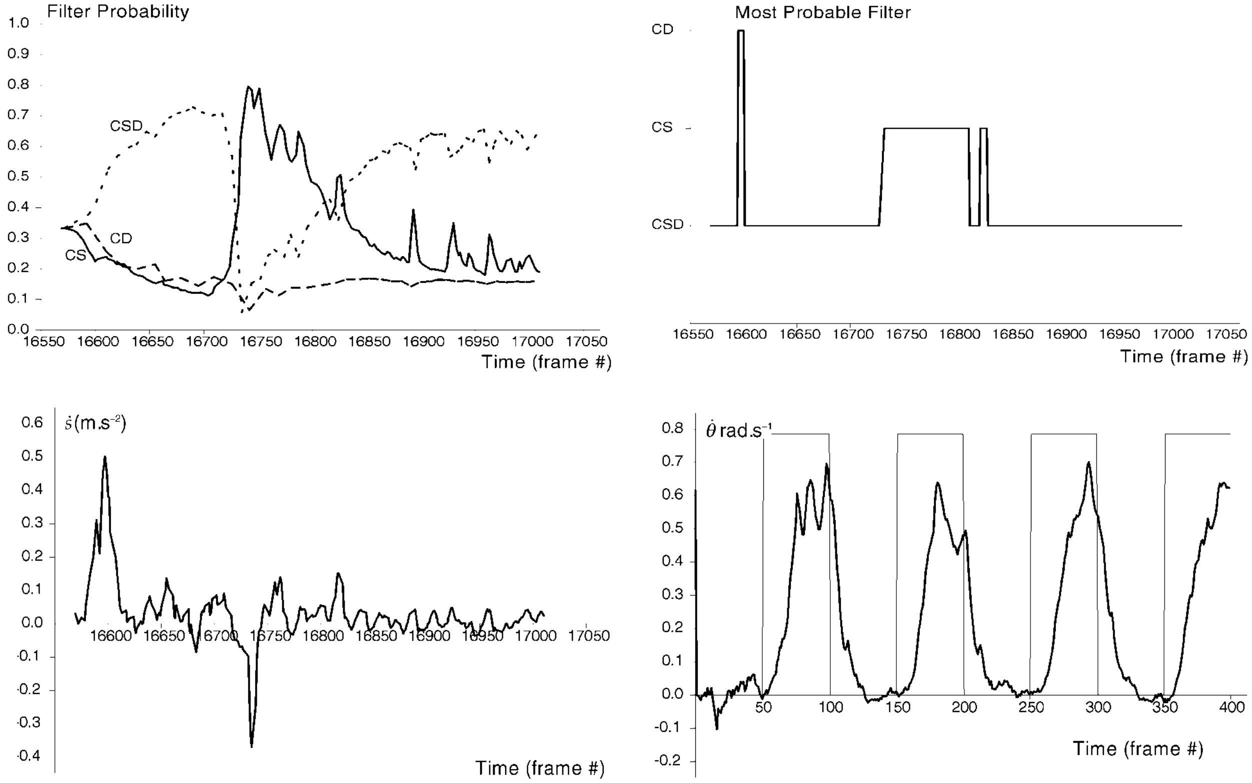


Fig. 13. IMM output for person trajectory. (a) Shows the filter likelihoods (where solid, dashed, and dotted lines are outputs from the CS, CD, and CSD filters). (b) Gives the most likely filter, where the dashed lines show when the actual maneuvers took place. (c) and (d) Show the overall IMM estimates of acceleration $\dot{s}$ in ms^{-2} and turn rate $\dot{\theta}$ in rad.s^{-1}.

the previous time-step and the model switching probabilities. The long-term performance of all filters is maintained as the filters with incorrect models at any time do not deviate significantly from the correct state vector, since mixing incorporates a large fraction of the correct state vector via the large filter likelihood $p_k(i)$ for the correct model i at step k.

The model switching probabilities are coded as $\mathbf{J}_k$, where $J_k(ij)$ represents the likelihood of switching from model i to model j at time-step k. As our expected targets are likely to continue with a single motion model for extended periods with only occasional model switching, we set the diagonal elements of the matrix as $J_k(mm) = J_d \approx 1$ and have small off-diagonal elements $J_k(mn) = (1 - J_d)/(N - 1)$, where $N = 3$ in our case. The filter likelihoods are initialized as $p_0(\mathrm{CSD}) = p_0(\mathrm{CS}) = p_0(\mathrm{CD}) = 1/3$. Fig. 11 illustrates the effect upon the model probabilities for the square trajectory of varying the switching probability from $J_d = 0.93$ to $J_d = 0.99$. The left hand figures (Fig. 11a1 and Fig. 11b1) illustrate the model probability sequence for the trajectory, the right hand figures (Fig. 11a2, and Fig. 11b2) the estimated turning rate values. In each figure, for the first 50 or so observations the filters exhibit similar performance to the above example of straight line motion. As the maneuver begins the likelihood of the CS filter rises and the CSD and CD filter likelihoods decrease. Around step 60 the CS filter becomes the most likely. During the maneuver the filter probabilities remain roughly constant, and when the maneuver completes the filter likelihoods adjust until the CSD filter again becomes most likely. The pattern continues as the state sequence repeats a further three times.

It can be seen that increasing J_d gives better discrimination between models as the filter probabilities are more widely separated. This is as we would expect, since increasing p reduces the amount of mixing between models, and the correct filter will obtain a more accurate estimate of the target state vector and attain a higher probability. However, increasing J_d reduces the sharpness of response of the filter, as we see from the left hand graphs. The lag of the estimated turning rate is increased, since increasing J_d reduces the amount of mixing between filters, and the time taken after model switching for the new filter to affect the parameter estimates is thus increased. Empirically, a value of $J_d = 0.96$ proved a satisfactory compromise.

Fig. 12 shows results for the square trajectory when $J_d = 0.96$. In the CSD state the mean values of $\dot{s}$ and $\dot{\theta}$ are small, confirming that the target is not maneuvering, whilst in the CS state we see mean values of $\dot{\theta} \approx 0.6$ rad s^{-1}, the actual value being 0.785 rad s^{-1}. It can be seen from Table 1 that the IMM detects changes in the motion model reliably, the correct filter being selected within ±10 observations. This error is not so important as far as the filtering required for improved target pursuit is concerned, as the position and velocity estimates which generate controller demands are consistent throughout the switching periods because of mixing. The important point for a higher level motion classification scheme is a reliable estimate of the state duration between switches and the mean acceleration and turning rate values.

Fig. 13 illustrates IMM results for the tracking of a walk-

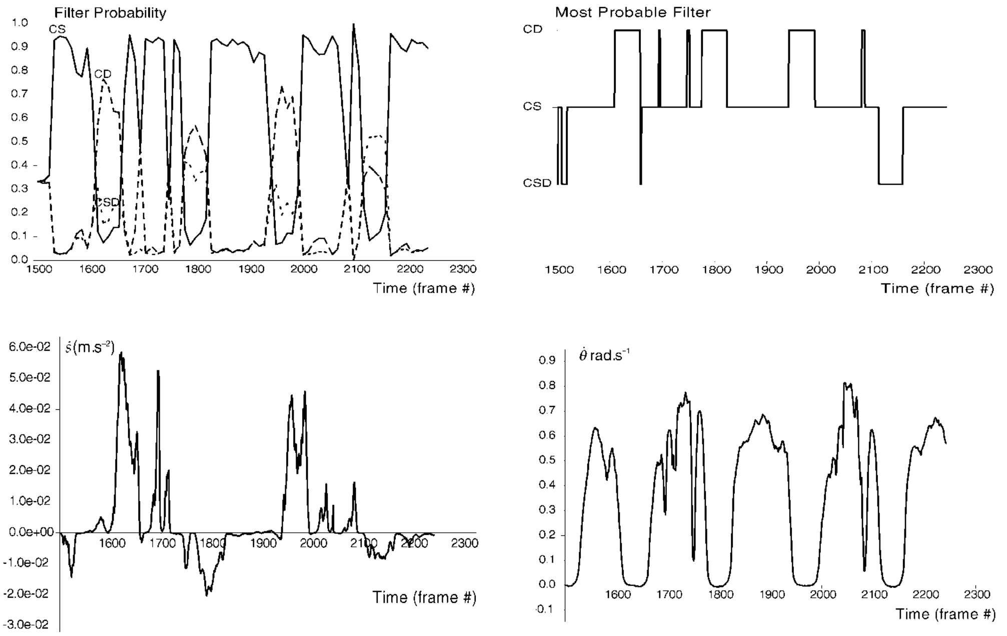


Fig. 14. IMM output for train in oval trajectory. (a) Shows the filter likelihoods (where solid, dashed, and dotted lines are outputs from the CS, CD, and CSD filters). (b) Gives the most likely filter, where the dashed lines show when the actual maneuvers took place. (c) and (d) Show the overall IMM estimates of acceleration $\dot{s}$ inms^{-2} and turn rate $\dot{\theta}$ in rad.s^{-1}.

ing person following the trajectory given in Fig. 10a (also one of those shown in Fig. 3a). The filter probabilities illustrate a period of straight line motion with occasional periods of slight acceleration, followed by a turning maneuver where the CS filter has highest likelihood, followed by further straight line motion. Table 2 illustrates mean model probabilities, acceleration and turning rate for each motion period.

The above tests have shown the advantages afforded by embedding filters within the IMM, namely much improved long term performance when trajectories incorporate periods of model switching. We now describe real-time implementation of the filter scheme and illustrate improved tracking performance of the system when filtering is incorporated.

6 Real-Time Filter Implementation

The IMM filter has been implemented in real-time to evaluate its potential for driving tracking from 3D rather 2D prediction. The first step of the filter loop is to evaluate the current state of the target predicted by each of the incorporated filters. Motion estimates received by the high-level gaze controller from the foveal motion process are then transformed to the world coordinate frame and the individual filter states updated according to the normal Kalman update step. The updated filter likelihoods are evaluated and the mixed output state vector calculated. The mixed state estimate of position and velocity are then converted to an angular position and velocity demands which are sent to the servo controller to drive platform motion.

The time taken to perform the prediction and update steps for each of the individual filters is 26ms, of which 19ms is a common overhead due to communication between processes and the storage and relay of results at each frame. Without mixing the total time taken to run predict and update steps for the three CSD, CS, and CD filters is 39ms, barely under the 40ms maximum permitted for operation at 25Hz. With mixing incorporated into the filter scheme the time taken rises to 53ms, and we are no longer able to run the complete filter cycle with mixing at 25Hz. To overcome this limitation we run the prediction step for each filter at each frame, but only update the filter states estimate with every second frame of received data, allowing the system to maintain 25Hz operation. Although such timings are processor dependent, this example illustrates that filter framework allows prediction not only over delays arising from latency, but also over delays due to missing data. Although direct timing comparisons between the IMM's three filters and a single image-based filter have not been performed, the above figures indicate that the computing time is increased less than four-fold.

Fig. 14 shows results from the IMM while the head platform tracks a toy engine which follows an oval made of two straights and two semi-circles (as recovered in Fig. 15). The filter probabilities show that around the corners the CS filter (filter 2) is selected and on straights periods of CSD and CD are mixed as the engine accelerates and decelerates. The periods of acceleration and four periods of turn (corresponding to two revolutions) are apparent in Fig. 14c and Fig. 14d. The total period of revolution was some 14 seconds, with some four seconds spent on the two sections of straight of total length 1.8m, and 10 seconds spent on the circular sections of total length 3.3m. The recovered value of $\dot{\theta} \sim 0.6 \mathrm{rad.s}^{-1}$ is close to the expected value of $2\pi/10$. The period of acceleration indicated in Fig. 14c occur along the straights, and indicate a change in speed from 0.4ms^{-1} to 0.5ms^{-1} along each straight section.

One of our key aims in this work was to demonstrate that prediction in 3D is more effective than that per-

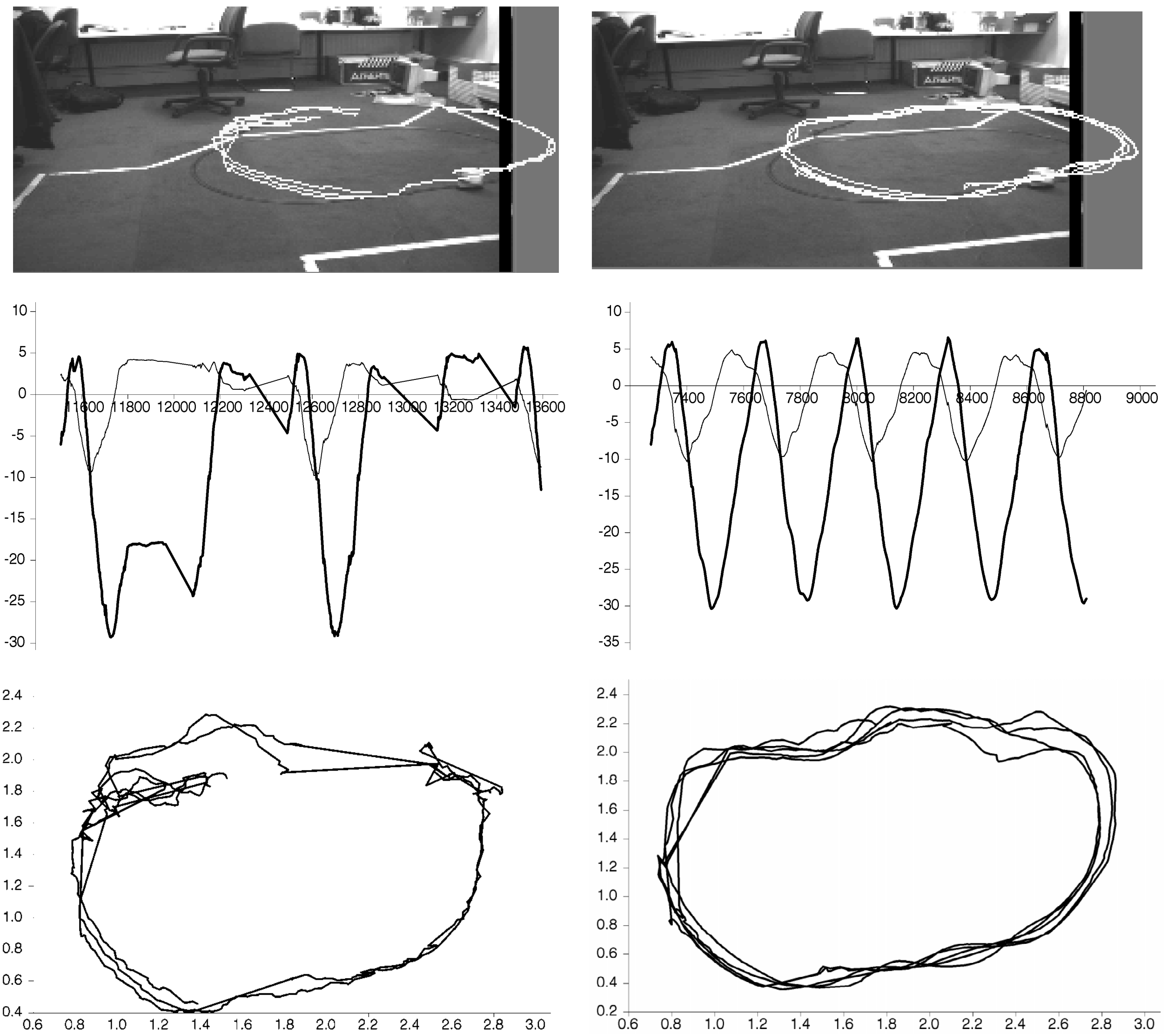


Fig. 15. The pursuit performance using 3D filtering and prediction contrasted with that from 2D filtering alone: (a) The trajectory mapped into reference image. (b) Vergence (thick line) and elevation (thin line) gaze angles over sequence. (c) The recovered ground plane trajectory. Figs. 15b-2D and 15c-2D show that, with 2D filtering alone, multiple saccades are required to recapture the target when pursuit breaks down.

formed in 2D. Fig. 15 illustrates the improvement when tracking the toy engine. The left sides of Fig. 15b and Fig. 15c show 2D tracking, where saccades have had to be resorted to in order to recapture the target when 2D pursuit fails. The right sides show the results when IMM filtering and 3D prediction is used. The system maintains pursuit without the use of corrective saccades.

7 Conclusions and Discussion

This paper has demonstrated first the tracking of targets moving in a plane in the scene using an active monocular camera and the recovery of "observer trajectories," the intersection of the instantaneous direction of gaze with the frontal plane. By calibrating the homography between the frontal plane and scene plane using plane to plane correspondences of at least four points or at least four lines, it was possible to reconstruct the scene trajectory of the target during tracking.

The paper then described the implementation of three Kalman filters based on simple canonical motions to filter the trajectory, and their embedding in an Interacting Multiple Model which allowed the most appropriate filter to describe the motion. An improvement in tracking performance was demonstrated in real time using the 3D trajectory and filter to provide prediction over the latency in the visual feedback loop, rather than using filtered 2D observer trajectories. The IMM was able to provided strong filtering and good prediction together with a robustness not available when using a single filter. In principle, there is no reason why the IMM method should not be extended to discriminate between fully 3D motions (recovered, say, from stereo tracking) rather than planar ones, although the addition of further state variables must increase uncertainty and hence make the distinction between filters more difficult within the IMM. In situations where the motion is not well described by any of the individual filters, the IMM is at best neutral, that is the latest datum is believed in preference to the filtered state. To what extent increasing the repertoire of simple motions and hence the number of filters is useful is open to question, but the early caveat about the competition between filters becoming more difficult must surely apply.

Although we have emphasized the use of filtering in 3D for prediction, an important issue in an active vision system is to what in the scene the system should devote its visual resources or attention. From this perspective, the detection of maneuvers, or the lack of them, is more relevant than prediction. An interesting target, and one which demands more resources, is one which maneuvers, whereas a boring one is one which does not. Our empirical evidence suggests that the Interacting Multiple Model provides not only strong filtering providing good prediction, but also a reliable—though inevitably delayed—indication of maneuver as the most likely model changes. The probability of changing model from moment to moment, which can be analyzed as a Markov chain [32], provides a characteristic motion signature for a target.

ACKNOWLEDGMENTS

This work was supported by EPSRC Grants GR/G30003 and GR/J65372, by an EPSRC Research Studentship to KJB, and by the Violette and Samuel Glasstone Research Fellowship in Science from the University of Oxford to IDR.

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 Vision," *Mechatronics*, vol. 3, no. 4, pp. 517–535, 1993.

[2] D.W. Murray, P.F. McLauchlan, I.D. Reid, and P.M. Sharkey, "Reactions to Peripheral Image Motion Using a Head/Eye Platform," *Proc. Fourth Int'l Conf. Computer Vision*, pp. 403–411, Berlin, 1993. Los Alamitos, Calif.: IEEE CS Press, 1993.

[3] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, and P.M. Sharkey, "Driving Saccade to Pursuit Using Image Motion," *Int'l J. Computer Vision*, vol. 16, no. 3, pp. 205–228, 1995.

[4] I.D. Reid and D.W. Murray, "Tracking Foveated Corner Clusters Using Affine Structure," *Proc. Fourth Int'l Conf. Computer Vision*, pp. 76–83, Berlin, 1993. Los Alamitos, Calif.: IEEE CS Press, 1993.

[5] I.D. Reid and D.W. Murray, "Active Tracking of Foveated Feature Clusters Using Affine Structure, "*Int'l J. Computer Vision*, vol. 18, no. 1, pp. 1–20, 1996.

[6] K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, and D.W. Murray, "Saccade and Pursuit on an Active Head/Eye Platform," *Image and Vision Computing*, vol. 12, no. 3, pp. 155–163, 1994.

[7] C.M. Brown, "Gaze Control with Interactions and Delays," *IEEE Trans. Systems, Man and Cybernetics*, vol. 63, pp. 61–70, 1990.

[8] C.M. Brown, "Prediction and Cooperation in Gaze Control," *Biological Cybernetics*, vol. 63, pp. 61–70, 1990.

[9] J.J. Clark and N.J. Ferrier, "Modal Control of an Attentive Vision System," *Proc. Second Int'l Conf. Computer Vision*, pp. 514–523, Tampa FL, 1988. Los Alamitos, Calif.: IEEE CS Press.

[10] P.M. Sharkey and D.W. Murray, "Coping With Delays for Real-Time Gaze Control," *Proc. SPIE Sensor Fusion VI*, Boston Mass., Sept. *1993*

[11] R. H. S. Carpenter, *Movements of the Eyes*. London: Pion, 1988.

[12] S.M. Fairley, I.D. Reid, and D.W. Murray, "Transfer of Fixation for an Active Stereo Platform via Affine Structure Recovery," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 1,100–1,105, Cambridge Mass., 1995. Los Alamitos, Calif.: IEEE CS Press, 1995.

[13] T.N. Tan, G.D. Sullivan, and K.D. Baker, "Structure From Motion Using Ground Plane Constraint," *Proc. Second European Conf. Computer Vision*, pp. 277–281, Santa Margherita, Italy, 1992. Heidelberg: Springer-Verlag.

[14] G.D. Sullivan, "Visual Interpretation of Known Objects in Constrained Scenes," *Phil. Trans. Royal Soc. London*, vol. B337, pp. 109–118, 1992.

[15] M. Mohnhaupt and B. Neumann, "Understanding Object Motion: Recognition, Learning, and Spatiotemporal Reasoning," *Robotics and Autonomous Systems*, vol. 8, pp. 65–91, 1991.

[16] D. Koller, K. Danilidis, T. Thòrhallson, and H.-H. Nagel, "Model-Based Object Tracking in Traffic Scenes," *Proc. Second European Conf. Computer Vision*, pp. 437–452, Santa Margherita, Italy, 1992. Heidelberg: Springer-Verlag.

[17] D. Koller, D. Danilidis, and H.-H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int'l J. Computer Vision*, vol. 10, no. 3, pp. 257–281, 1993.

[18] S.S. Intille and A.F. Bobick, "Closed-World Tracking," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 672–678, Cambridge Mass., 1995. Los Alamitos, Calif.: IEEE CS Press, 1995.

[19] H.A.K. Blom, "An Efficient Filter for Abruptly Changing Systems," *Proc. 23rd IEEE Conf. Decision Control*, pp. 656–658, 1984.

[20] K. Pahlavan and J.-O. Eklundh, "A Head-Eye System-Analysis and Design," *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 41–56, 1992.

[21] K. Pahlavan, T. Uhlin, and J.-O. Eklundh, "Dynamic Fixation," *Proc. Fourth Int'l Conf. Computer Vision*, pp. 412–419, Berlin, 1993. Los Alamitos Calif.: IEEE CS Press, 1993.

[22] A. Blake, R. Curwen, and A. Zisserman, "A Framework for Spatiotemporal Control in the Tracking of Visual Contours," *Int'l J. Computer Vision*, vol. 11, no. 2, pp. 127–146, 1993.

[23] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[24] J.L. Mundy and A.P. Zisserman, eds., *Geometric Invariance in Computer Vision*. Cambridge Mass.: MIT Press, 1992.

[25] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[26] G.A. Watson and W.D. Blair, "IMM Algorithm for Tracking Targets that Maneuver Through Coordinated Turns," *SPIE Proc. Signal and Data Processing of Small Targets*, SPIE vol. 1,698, pp. 236–247, 1992.

[27] G.A. Watson and W.D. Blair, "The IMM Algorithm and Aperiodic Data," *SPIE Proc. Acquisition, Tracking and Pointing VI*, SPIE vol. 1,697, pp. 83–91, 1992.

[28] Y. Bar-Shalom, K.C. Chang, and H.A. Blom, "Tracking a Maneuvering Target Using Input Estimation Versus the Interacting Multiple Model Algorithm," *IEEE Trans. Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 296–300, 1989.

[29] C-B. Chang and J.A. Tabaczynski, "Application of State Estimation to Target Tracking," *IEEE Trans. Automatic Control*, vol. 29, no. 2, pp. 98–109, 1984.

[30] P. Andersson, "Adaptive Forgetting in Recursive Identification Through Multiple Models," *Int'l J. Control*, vol. 42, no. 5, pp. 1,175–1,193, 1985.

[31] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo Calif.: Morgan Kauffman, 1988.

[32] K.J. Bradshaw, *Active Visual Surveillance*, DPhil Thesis, Univ. of Oxford, 1995.

Kevin J. Bradshaw graduated with first class honors in physics in 1991 from Heriot-Watt University, Edinburgh, and received a DPhil in engineering science in 1995 from the University of Oxford. His doctoral work was concerned with the use of a closed-loop active vision system to perform automated tracking within dynamic scenes, part of which is summarized in this paper. Since 1995 he has worked with Smith System Engineering, a UK-based science and technology consultancy, as a consultant in the Environment and Science Policy Sector.

Ian D. Reid received the BSc with first class honors in computer science from the University of Western Australia in 1987 and was awarded the Western Australian Rhodes Scholarship for that year. He completed a DPhil in engineering science at the University of Oxford in 1991 on the subject of recognizing parametric models in range data. Since then he has been a postdoctoral researcher in engineering science investigating the theory and practice of uncalibrated active vision for tracking and navigation and has published over 30 papers on these and other topics. He currently holds a Junior Research Fellowship (The Queen's College) and a Violette and Samuel Glasstone Research Fellowship in Science at the University of Oxford.

David W. Murray received a BA with first class honors in physics in 1977 and a DPhil in physics in 1980, both from the University of Oxford. After a period as a research fellow at California Institute of Technology and as a staff scientist at the GEC Research Laboratories in London, he returned to Oxford in 1989 as a university lecturer in the Department of Engineering Science. Since 1983 his research interests have centered on the computation of image motion and structure from motion, more recently applying these to active and teleoperated visual systems. He has published some 90 papers in physics and machine vision and coauthored, with Bernard Buxton, a book, *Experiments in the Computation of Visual Motion*, MIT Press, 1990. He holds a Tutorial Fellowship in Engineering Science at St. Anne's College, Oxford.