Learning Surveillance Tracking Models for the Self-Calibrated Ground Plane

J. Renno, J. Orwell and G.A. Jones
Digital Imaging Research Centre, Kingston University,
Penrhyn Road, Kingston upon Thames, Surrey, UK, KT1 2EE
{j.renno,j.orwell,g.jones}@kingston.ac.uk

Abstract

Tracking strategies usually employ *motion* and *appearance* models to locate observations of the tracked object in successive frames. The subsequent model update procedure renders the approach highly sensitive to the inevitable observation and occlusion noise processes. In this work, two robust mechanisms are proposed which rely on knowledge about the ground plane. First a highly constrained bounding box appearance model is proposed which is determined solely from predicted image location and visual motion. Second, tracking is performed on the ground plane enabling global real-world observation and dynamic noise models to be defined. Finally, a novel auto-calibration procedure is developed to recover the image to ground plane homography by simply accumulating event observations.

1 Introduction

By far the most common approach to tracking in typical surveillance imagery uses *pixel differencing* and *blob analysis*. Typically motion detection extracts moving regions from static scenes[7]. *Trajectory tracking* is employed to establish the temporal history of individual objects. An iterative estimator (*e.g.* Kalman or α-β) is employed to update a first or second order visual *trajectory model*. Temporal correspondence (or *data association*) is achieved essentially using simple Newtonian physics either locally for each object, or globally by considering all possible object-observation pairings[1]. Additionally an *appearance model* matching may be employed to improve tracking accuracy by comparing width and height, shape or colour[6, 5]. While surprising successful, maintaining temporal correspondence is a significant problem particularly through occlusion and fragmentation where the shape, dimensions and colour signature of the merged or fragmented observations do not correlate well with the actual object, or where the trajectory model does not correspond to actual object trajectory.

Two related problems are addressed in this paper. First the problem of frequent fragmentation and merging of moving regions caused by occlusion and low contrast processes. These unexpected regions usually introduce considerable noise into the *data association* phase of the tracker and, more subtlely, into the updating of the trajectory and appearance models which is then propagated into the subsequent frames. Typical solutions are complex and *ad hoc* split and merge procedures applied to observation and appearance model primitives[3, 4, 2].

BMVC 2002 doi:10.5244/C.16.59

The second problem relates to the choice of motion model. Linear pixel-based motion models applied to trajectory and appearance models are too constrained to adequately model the evolution of objects - particularly vehicles. The result is frequent loss of correpondence as objects manoeuvre in the scene. On the other hand, more appropriate quadratic models are easily mislead by observation noise. The difficulty lies in the problem of establishing global pixel-based noise models which are appropriate to both distant visually-slow objects at the top of the image, and closer objects with larger visual velocities at the bottom.

In this work, we introduce three mechanisms to address these problems which rely on knowledge about the ground plane. First we develop highly discriminatory bounding-box appearance models of scene objects which indirectly use the depth of the object to model its projected width and height. Since, the spatial extent of an object is now a function of image position, the tracker will be more robust when presented with the distorted observations which arise from fragmentation or occlusion processes. Second, the observations are transformed onto the ground plane coordinate system within which a quadratic rather than linear motion model is defined. Global real-valued noise models can be generated for observation and dynamic noise models. Finally, rather than relying on a labour-intensive calibration procedures to recover the image to ground-plane homography[8], the system relies on a simple auto-calibration procedure to learn the relationship between image and world by simply watching events within the monitored scene.

2 Auto-Calibration of the Ground Plane

In this section a simple yet highly effective method of learning the image to ground plane homography of the camera is presented which exploits the simple but reasonably accurate assumption that in typical surveillance installations, *the projected 2D image height of an object varies linearly with its vertical position in the image* - increasing down the image from zero at the horizon. This height model is derived from the optical geometry of a typical visual surveillance installation. In addition, such an assumption enables the use of simple but highly discriminatory models of the appearance of scene objects which indirectly use the depth of the object to model its projected height. In this *auto-calibration* scenario, the *ground plane coordinate system* (GPCS) is defined as follows:

The Y-axis $\hat{\boldsymbol{Y}}$ of the GPCS is defined as the projection of the optical axis along the ground plane. The Z-axis $\hat{\boldsymbol{Z}}$ is defined as the ground plane normal. The position of the camera focal point in the GPCS is 'above' the GPCS origin at the point $(0, 0, L)$.

2.1 Ground Plane Projection

The image plane is situated at distance f (focal length of the optical system for the camera) perpendicular to the optical axis $\hat{z}$. In this configuration a point P on the image plane has coordinates $\boldsymbol{x}' = (x, y, -f)^T$. The pixel coordinate system i, j (representing the row and column position respectively) is related to the image plane coordinate system by $x = \alpha_x(j - j_0)$ and $y = \alpha_y(i_0 - i)$ where i_0, j_0 is the optical centre of the image and α_x and α_y are the horizontal and vertical inter-pixel widths. Thus $\boldsymbol{x}' = (\alpha_x^f(j - j_0), \alpha_y^f(i_0 - i), -1)^T f$ where α_x^f and α_y^f are the horizontal and vertical pixel dimensions normalised by the focal length.

An optical ray containing the focal point of the camera passing through the image plane can be represented in vectorial form as $\boldsymbol{x} = \mu \boldsymbol{x}'$. Let Q be the point of intersection of the optical ray with the ground plane Π. In order to calculate the position of the point Q on the ground plane Π in the ground plane coordinate system, one must convert the position of a point given the transformation $(\boldsymbol{R}, \boldsymbol{t})$ between the image plane and world coordinate systems *i.e.* $\boldsymbol{X} = \mu \boldsymbol{R}\boldsymbol{x}' + \boldsymbol{t}$. Writing the ground plane equation as $\boldsymbol{n}_\Pi \cdot \boldsymbol{X} = 0$, where the ground plane normal $\boldsymbol{n}_\Pi \equiv \hat{\boldsymbol{Z}}$, then the position $\boldsymbol{X}$ of the point Q is obtained by noting that $\boldsymbol{X} \cdot \hat{\boldsymbol{Z}} = 0$.

$$\mu = -t_z / \hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}' \tag{1}$$

The local GPCS is defined with a zero pan angle. Assuming no significant roll angle, then after some algebraic manipulation, the ground plane coordinates may be related to the look-down angle θ as follows

$$\frac{X}{L} = \frac{\alpha_x^f \, (j - j_0)}{\alpha_y^f \, (i - i_0) \sin\theta - \cos\theta}, \quad \frac{Y}{L} = -\frac{\alpha_y^f \, (i - i_0) \cos\theta - \sin\theta}{\alpha_y^f \, (i - i_0) \sin\theta - \cos\theta} \tag{2}$$

Thus to compute the ground plane position of an image point, the following camera parameters i_0, j_0, α_x^f, α_y^f and θ are needed. In our approach the optical centre i_0, j_0 is computed by an optical flow algorithm which robustly fits a global zoom motion model to a three frame sequence undergoing a small zoom motion.

2.2 Projected Object Height

If one assumes that the height of a moving object is known (*i.e.* a person) then the point of intersection $\boldsymbol{X}$ with the ground plane can be shifted along the $\hat{\boldsymbol{Z}}$ direction by the height H. Using μ, we can write $\boldsymbol{X}' = \mu \boldsymbol{R}\boldsymbol{x}' + \boldsymbol{t} + H\hat{\boldsymbol{Z}}$. The new image point $\boldsymbol{x}''$ corresponding to the projection of the top of the person can be computed from the inverse transformation $\boldsymbol{R}^T(\boldsymbol{X}' - \boldsymbol{t})$ to yield

$$\lambda \boldsymbol{x}'' = \mu \boldsymbol{x}' + H\boldsymbol{R}^T \hat{\boldsymbol{Z}} \tag{3}$$

where λ is the projection factor from the image plane to the top of the person. Substituting μ from equation 1 and $t_z = L$ yields

$$\boldsymbol{x}'' = -\frac{1}{\lambda}\left(H\boldsymbol{R}^T\hat{\boldsymbol{Z}} - \frac{L}{\hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}'}\boldsymbol{x}'\right) \tag{4}$$

To measure the projected vertical height of an object, we simply define a plane Λ containing the optical centre and the image plane raster line containing the new point $\boldsymbol{x}''$. The

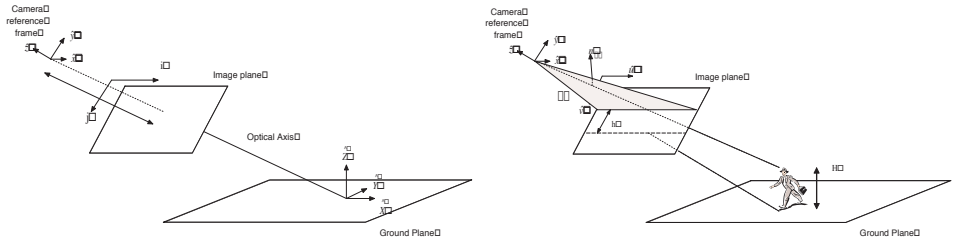

Figure 1: (a) Camera, World and Pixel Coordinate Systems (b) Projected Height

normal $\boldsymbol{n}_\Lambda$ of this plane is defined by the cross-product between the projection line $\lambda \boldsymbol{x}''$ and the rasterline direction vector $\hat{\boldsymbol{x}}$ as follows

$$\boldsymbol{n}_\Lambda = -\frac{1}{\lambda}\left(H\boldsymbol{R}^T\hat{\boldsymbol{Z}} \times \hat{\boldsymbol{x}} - \frac{L}{\hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}'}\boldsymbol{x}' \times \hat{\boldsymbol{x}}\right) \tag{5}$$

The rasterline containing the point $\boldsymbol{x}''$ can be thought of as lying at a distance h above the projection of the bottom of the person - see Figure 1(b). Therefore the point vertically above $\boldsymbol{x}'$ can be expressed as $\boldsymbol{x} = \boldsymbol{x}' + h\hat{\boldsymbol{y}}$ and belongs to the plane Λ. Substituting $\boldsymbol{x}' + h\hat{\boldsymbol{y}}$ into the equation of plane Λ, $\boldsymbol{n}_\Lambda \cdot \boldsymbol{x} = 0$, generates

$$h = -\frac{\boldsymbol{n}_\Lambda \cdot \boldsymbol{x}'}{\boldsymbol{n}_\Lambda \cdot \hat{\boldsymbol{y}}} \tag{6}$$

Further simplification can be derived by expanding the numerator and denominator of equation 6 using equation 5 as follows

$$-\lambda \boldsymbol{n}_\Lambda \cdot \boldsymbol{x}' = H(\boldsymbol{R}^T\hat{\boldsymbol{Z}} \times \hat{\boldsymbol{x}}) \cdot \boldsymbol{x}' - \frac{L}{\hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}'}(\boldsymbol{x}' \times \hat{\boldsymbol{x}}) \cdot \boldsymbol{x}' \tag{7}$$

$$= H(\boldsymbol{R}^T\hat{\boldsymbol{Z}} \times \hat{\boldsymbol{x}}) \cdot \boldsymbol{x}' \tag{8}$$

since $(\boldsymbol{x}' \times \hat{\boldsymbol{x}}) \cdot \boldsymbol{x}' = 0$, and

$$-\lambda \boldsymbol{n}_\Lambda \cdot \hat{\boldsymbol{y}} = H(\boldsymbol{R}^T\hat{\boldsymbol{Z}} \times \hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{y}} - \frac{L}{\hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}'}(\boldsymbol{x}' \times \hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{y}}$$

$$= H(\boldsymbol{R}^T\hat{\boldsymbol{Z}} \times \hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{y}} - \frac{Lf}{\hat{\boldsymbol{Z}} \cdot \boldsymbol{R}\boldsymbol{x}'} \tag{9}$$

where $\boldsymbol{x}' \times \hat{\boldsymbol{x}}) \cdot \hat{\boldsymbol{y}} = f$. Where there is a zero roll angle, equations 8 and 9 combine to generate the following expression for image plane height h which depends only on object height H, camera height L and vertical image height y.

$$h = \frac{(f^2 - y^2)\sin\theta\cos\theta + yf(\cos^2\theta - \sin^2\theta)}{y\sin\theta\cos\theta - (\cos^2\theta - L/H)f} \tag{10}$$

For typical camera installations, h can be shown to effectively vary linearly with vertical image position relative to the position of horizon. Figure 2 plots projected height against image position for a typical 512×512 camera with focal length $f = 10mm$ and angular field of view of $\approx 33°$. Three different camera height/lookdown angle configurations are shown $L = 6m/\theta = -76$ (*i.e.* $14°$ down from horizontal), $L = 5m/\theta = -80$ and $L = 15m/\theta = -65$. Average person height is assumed to be $H = 1.76m$. Note for the given range of image positions, the plot is essentially linear. The intercept with the vertical position axis (or $h = 0$ axis) defines the horizon where objects become infinitely small.

Such a linear model may be extracted from the scene automatically by accumulating a histogram $H[i, \mu]$ (where μ is the pixel height) from a large number of detected moving regions of the monitored scene - see figure 3[1] and 4. Currently the operator drags a line segment along the ridge structure to define the gradient γ and horizon i_h.

[1] PETS 2001 Datasets `visualsurveillance.org`

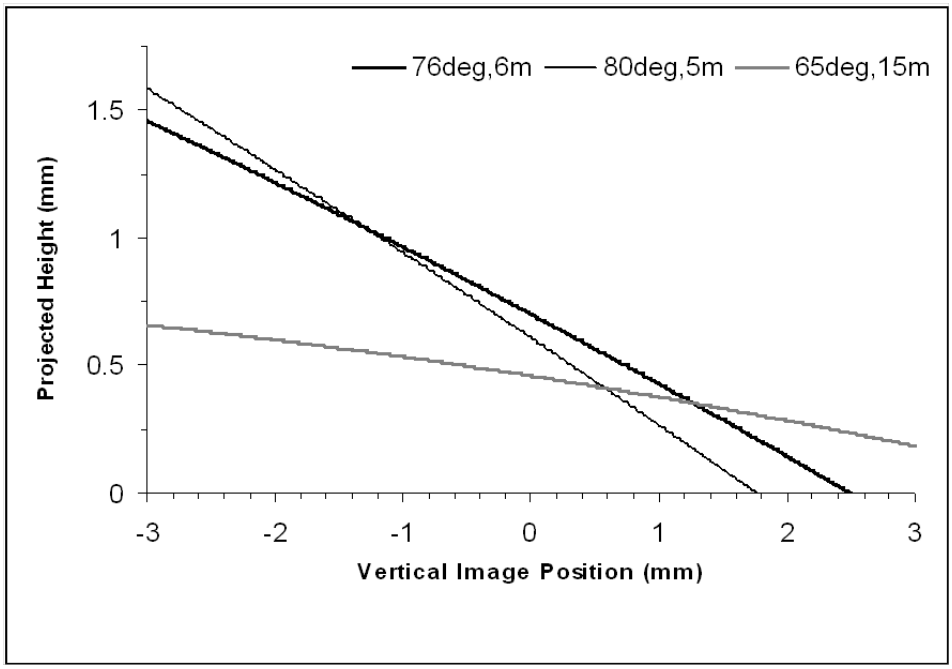

Figure 2: Projected Height versus Vertical Position

Figure 3: (a) PETS Camera 1 (b) PETS Camera 2 (c) Football

Figure 4: (a) DIRC Camera 1 (b) DIRC Camera 2 (c) DIRC Camera 3

2.3 Ground Plane Calibration

Since the vertical image height of an object is independent of the horizontal image position of the projected object, the following derivation may assume, without loss of generality, that the object is located on the vertical axis *i.e.* $x = 0$. Two key positions of a projected object may be defined at $i = i_h$ at the horizon, and $i = i_0$ at the optical centre of the image. At the former, the look-down angle θ may directly related to the horizon parameter i_h extracted from the accumulated training data acquired in the learning stage described in section 2.2 *i.e.*

$$\cot\theta = \alpha_y^f (i_h - i_0) \tag{11}$$

For the latter case, consider an object of height H standing on the ground plane point given by the projection of the optical axis. From equation 10, the vertical height at this point $h(i = i_0)$ may be related to the look-down angle as follows

$$\frac{h}{f} = \frac{H\cos\theta\sin\theta}{L - H\cos^2\theta} \tag{12}$$

An estimate of the height h may also be generated from the learnt linear projected height model *i.e.* $h(i_0) = \alpha_x\gamma(i_0 - i_h)$. Combining this with equations 11 and 12, the following expressions for the camera parameters θ and α_y^f may be derived

$$\sin^2\theta = \frac{\gamma}{H}\frac{L-H}{1-\gamma}, \quad \alpha_y^f = \frac{\cot\theta}{(i_0 - i_h)} \tag{13}$$

2.4 Evaluating Ground Plane Calibration

The following section evaluates the accuracy of the auto-calibration technique by comparing the estimated look-down angle with that recovered using the standard Tsai calibration procedure[8]. The Tsai calibration results performed on the PETS2001 [2] were not particularly accurate at estimating the camera height and look-down angle. Consequently the evaluation was performed on four datasets; three local installations (DIRC 1, 2 and 3), and a Football dataset. The DIRC test installations involve different types of camera placed at different heights overlooking a common carpark scene. The carpark has been surveyed to generated real-world ground plane positions in a common coordinate system. These points have been selected to ensure that each camera has ten well distributed point in the image plane. The convex hull of these points contains most of the carpark and over fifty percent of the visual plane. The real lookdown angles and camera heights have been established using surveying equipment from the ground plane projection of the correct optical axes. Results are shown in Table 1. In all cases, the accuracy of the Tsai method and our own is comparable, with the shallow angle of view being the most problematic. Additionally the Tsai confirmed that the camera had no significant roll - typically less than $4°$.

[2] The PETS2001 datasets (visualsurveillance.org) are problematic as they contain so few tightly distributed calibration points.

Test Installations	DIRC 1	DIRC 2	DIRC 3	Football
Correct Angle	$16.0°$	$24.3°$	$13.5°$	n/a
Tsai Angle	$16.7°$	$24.5°$	$7.7°$	$23.0°$
Our Approach	$15.5°$	$23.3°$	$11.7°$	$26.2°$

Table 1: Look-Down Angle *For clarity the look-down angle has been redefined as $\pi/2-\theta$ - the angle of intersection between ground plane and optical axis.*

(a) Typical object (b) Detected pixels (c) Object Model

Figure 5: Modelling detected events: Images show (a) a typical example of a scene object, (b) the pixels detected as moving, and (c) the Object Model.

3 Model-based Tracking

In this section, the projected height concept is employed to define simple yet highly effective bounding box appearance models for the principle object types within a surveillance scene. The representation is composed of two vertically adjacent connected bounding boxes - the *object component* and *base component*. The *base* is the large number of background pixels beneath an object and the shadow regions which are typically segmented with the object pixels themselves. The object component is defined by (i) the vertical extent of the object - the *height model*, (ii) the horizontal extent of the object - the *width model*, and (iii) the vertical extent of the base region - the *base model*. These models, as illustrated in figure 5, are defined relative to the 2D position of the object - the 2D projection of the position of the object on the ground plane. Three different models are currently used corresponding to each of the principle vehicles types ψ in the set $\Psi = \{Person,\ Vehicle,\ Large\,Vehicle\}$. As with the ground-plane auto-calibration, the parameters for each of these models must be computed in a learning procedure.

$$\begin{aligned} \mu &= \Gamma^{\psi}\,(i - i_h) \\ \omega &= \Omega^{\psi}(\theta)\,(i - i_h) \\ \beta &= \beta^{\psi}(\kappa)\Gamma^{\psi}\,(i - i_h) \end{aligned} \tag{14}$$

The Height Model: The expected pixel height μ (see equation 14) varies linearly with vertical image position i. Different height models $\Gamma^{\psi},\ \psi \in \Psi$ must be defined for each type of object ψ - see figure 6(a). A further assumption is made that the projected height of vehicles does not depend on the orientation of the object.

The Width Model: For vehicular objects, the projected pixel width ω varies both as a function of depth (and hence varies linearly with position i) but also varies as a function

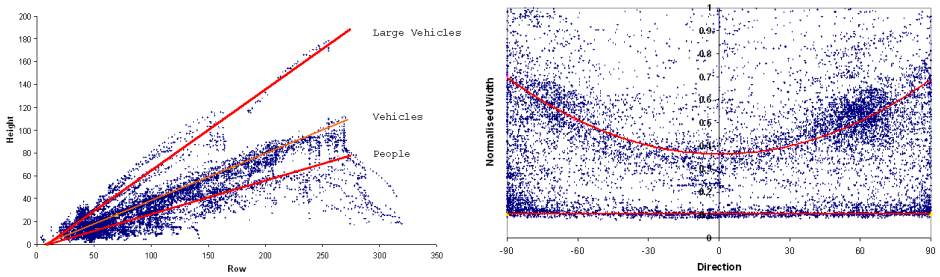

Figure 6: (a) Projected Height (b) Normalised Width. $\theta = 0$ *indicates vertical motion, while* $\theta = \pm\pi/2$ *refers to horizontal motion. The lower plot illustrates that* Person *width does not depend on orientation. For vehicles, the width increases from a minimum at* $\theta = 0$ *(*face-on*) to a maximum at* $\theta = \pm\pi/2$ *(*side views*).*

of the 3D orientation of the object. The 3D orientation of a moving vehicle is correlated with the direction of its visual motion. This relationship can be clearly demonstrated in 6(b) which plots 2D width (normalised by vertical height) against the visual motion direction θ for a large set of detected regions. Thus the projected width of an event is a function both i and the direction θ.

The Base Model: The vertical extent of the base again varies linearly with the vertical image position. In dull weather conditions, this base area is usually a small fraction. However in bright weather conditions, this base area can be become significantly larger. Currently, the base model parameter β^{ψ} is manually set as a proportion of the height model. Ideally some environmental illumination parameter κ would select the appropriate ratio.

4 Results

The *Ground Plane Tracker* (GPT) embeds the mechanisms introduced in this paper within a standard tracking framework, and is compared against a standard 2D tracker - the *Image Plane Tracker* (IPT). Both mechanisms employ a Kalman filter model whose observation and dynamic noise models are learnt directly from the data. The two methods are summarized in Table 2 below. Data association is performed by searching predicted bound-

Algorithm	Image Plane Tracker	Ground Plane Tracker
Measurements	x, y image pixels	X, Y ground plane - equation 2
Motion Model	First-order $x, y, \dot{x}, \dot{y}$	Second-order $X, Y, \dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$
Appearance Model	First-order Kalman filter on bounding box dimensions $h, w, \dot{h}, \dot{w}$	Position and velocity constrained bounding box model of equation 14

Table 2: Implementational Details of Standard and Proposed Tracking Algorithms

ing boxes for union of overlapping moving regions whose area is greater than 10% of bounding box area. Model instances are instantiated from unassigned *moving regions*[7]

whose areas are greater than some common threshold - 10 pixels (in quarter-size PAL frames). In neither case is any additional appearance matching implemented to improve data association. Observation position error is defined as deviation from predicted object dimension. Each object has a *time-to-live* counter (TTL) defined as $\min(TTL, 10)$ which is incremented if inter-frame match recovered, and decremented if no match recovered with object deleted when $TTL < 0$.

To compare the different approaches a *tracking error* is defined as the number of *track failures* per 1000 *track frames*. A track failure occurs when the tracking identity of any ground truth object changes. Track frames are the total number of object appearances for all tracks in a sequence. The experiment is run on three different datasets - see Figure 2: the PETS 2001 Dataset 1[3] (an occlusion rich dataset of distant objects in good lighting conditions), the Kingston Carpark Dataset (although relatively free of non-static occlusions, objects exhibit considerable motion variation against background undergoing frequent and severe lighting variations caused by intermittent direct and reflected sunshine), and the Football Dataset (large number of objects undergoing correlated and rapidly changing motions). Note that the tracking results reflect the challenging nature of

Tracker	PETS	DIRC	Football
IPT	3.2	1.5	49
GPT	1.9	1.1	11

Table 3: Tracking Error

the Kingston datasets and, in particular, the Football Dataset. Nonetheless, the proposed tracker outperforms the traditional tracker which is easily misled. Greater insight into the problems of trackers can be gained by determining the nature and frequency [4] (% of frames) of the failure modes - see Table 4. Both trackers loose track of objects that are

Failure	Description of data association failure	Frequency	
		IPT	GPT
Fragmentation	Unexpected small displaced observation	9%	2%
Static Occlusion	Unexpected small displaced observation	23%	10%
Object Occlusion	Unexpectedly large observation	36%	34%
Motion Model	Motion model too constraining	21%	34%
Stationary Object	Object merges into background	11%	20%

Table 4: Tracking Error

stationary for several seconds - determined by a TTL parameter. However the principal weakness of the traditional tracker is when dealing with situations where (i) fragmentation or static occlusion processes shrink the search window with consequent failure to locate validating observations, and (ii) occlusions which widen the search window causing the tracker to be deflected by the occluding object. These problems are more likely in situations where the trajectory deviates from the assumed motion model.

[3] `visualsurveillance.org/PETS2001/`

[4] Frequency will be highly dependent on dataset.

5 Conclusions

Traditional trackers employ simple bounding box search windows (a rudimentry *Appearance Model*) to search for an appropriately sized moving region to validate the tracked object. These search windows are typically smoothed height and width dimensions either using $\alpha - \beta$ filters or Kalman filters. When used in parallel with trajectory tracking, these search windows are often unstable leading to rapid track loss in noise, static or object occlusion. This is partly due to the ease in which the dimensions of the search window are distorted by the presence of fragmented or occluded moving regions, and partly due to the typically overconstrained linear trajectory models employed. Second-order motioon models are particularly difficult to control in image plane coordinates as even constant ground plane motion is inappropriately modelled on the projected image.

One approach is to embed more complex (and computer intensive) appearance models into the tracking framework. However, this commendable approach may be complemented by employing the highly constrained model-driven bounding box search window proposed here, and can dramatically and demonstrably (see Table 3) improve the temporal coherence of the underlying tracking process.

References

[1] Y. Bar-Shalom and T. Fortmann. *"Tracking and Data Association"*. Mathematics in Science and Engineering. Academic Press, 1988.

[2] F. Cupillard, F. Bremond, and M. Thonnat. "Tracking Groups of People For Video Surveillance". In *2nd European Workshop on Advanced Video-based Surveillance Systems*, pages 88–100, Kingston, UK, September 4 2001.

[3] T. Ellis and M. Xu. "Object Detection and Tracking in an Open and Dynamic World". In *Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Hawaii, December 2001.

[4] L.M. Fuentes and S.A. Velastin. "People Tracking in Surveillance Applications". In *Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Hawaii, December 2001.

[5] I.Haritaoglu, D.Harwood, and L.S.Davis. "W4: Real-time Surveillance of people and their Activities". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):809–830, July 1997.

[6] N.T. Siebel and S.J. Maybank. "Real-time Tracking of Pedestrians and Vehicles". In *Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Hawaii, December 2001.

[7] C. Stauffer and W.E.L. Grimson. "Learning Patterns of Activity using Real-Time Tracking". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[8] Roger Y. Tsai. "A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses". *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, August 1987.