Tracking without Feature Detection

Arthur E.C. Pece, Anthony D. Worrall
Department of Computer Science
University of Reading
P.O. Box 225
Reading RG6 6AY, England
{A.E.C.Pece, Anthony.Worrall}@Reading.ac.uk

Abstract

The tracking method presented in this paper is based on an active-contour algorithm for pose refinement. The most important innovation with respect to most active-contour methods is that no "features" are detected at any stage: the goodness-of-fit measure for the model is a smooth function (without thresholds) of both model parameters and image grey-levels. This smoothness leads to an estimate of the covariance of the likelihood of the model parameters. This covariance estimate is used for efficient pose optimisation by a Newton-like method and for proper weighting of the innovation in a Kalman filter. The method is demonstrated by tracking motor vehicles with 3-D models.

1. Introduction

The kernel of the tracking system described in this paper is a pose refinement algorithm which is a development of previous work in our research group [4, 21]. To understand the concept behind the algorithm, it is best to put it into the context of other model-based methods. Several such methods (*e.g.* [6, 12, 13] operate in two stages: at the first stage, a set of features (for instance, edges or corners) is extracted from the image; at the second stage, some of these features are matched to model features by maximising some criterion of goodness-of-fit. The best match can be found either by a direct solution or by an iterative (*e.g.* Newton-Raphston) search algorithm. By establishing a finite set of features, the first stage turns the pose refinement problem into a combinatorial problem [6]. The disadvantages of two-stage methods originate from feature detection and the establishment of correspondences between model features and image features: a threshold must be set at the feature-detection stage; a low threshold leads to a large combinatorial search, while a high threshold leads to loss of information which might be useful for the search.

Active contours [3] overcome the above disadvantages to some extent. In this class of methods, the extraction of features from the image is guided by a hypothesis about the model parameters: this hypothesis is used to instantiate the model and the search for image features takes place in the neighbourhood of the model contours. By eliminating the first stage of processing, it becomes feasible to search for simpler features without the risk of a combinatorial explosion: instead of correspondences between edges or corners or even more complex features, active contours search for correspondences between points on edges. However, most active-contour methods still involve thresholding [2, 7, 10, 20, 23]. After thresholding, the correspondences between points on edges are often determined locally, by the strength of image edge-points and their proximity to model contours: this strategy can lead away from the globally-optimal match (this criticism does not apply to the Condensation algorithm [8]).

Even if the problems that arise in finding the right correspondences are solved, the thresholds necessary for feature detection inevitably make the above methods less robust against noise. The method described in this paper avoids feature extraction altogether, and therefore the correspondence problem is also avoided: the principle is still the optimisation of an evaluation function (or objective function) computed from grey levels in the neighbourhood of model contours, but no feature extraction, *i.e.* no thresholding, is involved in computing the evaluation function. The theory behind the method is closer to classical statistics than to Bayesian statistics: the evaluation function that is optimised is a measure of the probability of the null hypothesis, that the measurements arise from clutter, rather than from the object being tracked. The principle was developed in previous work [4, 21] and an earlier version of the method was presented in [15, 17, 16]. This paper shows that, under mild assumptions about the object being tracked (in fact, by assuming ignorance about the contrast of its edges), the evaluation function has a simple relationship to likelihood and

Bayesian evidence. Apart from its theoretical significance, this relationship allows us to combine the pose-refinement method with a Kalman filter for efficient tracking.

The image statistic, used to evaluate the goodness-of-fit of the model pose, is a smooth function both of the position of the contour and of the grey levels; as a consequence, it lends itself to gradient-based optimisation. An additional feature of this statistic is that it allows an estimate of the covariance of the likelihood of pose parameters, and therefore leads to a Newton-like method for pose optimisation and to proper weighting of the innovation in the Kalman filter.

A method based on similar principles [11] has the disadvantages of not having a statistical interpretation of the evaluation function being optimised, and of being computationally much more expensive.

1.1. Organisation of the paper

Section 2 describes the evaluation function for points on the model contours, for straight model line-segments and for the entire model; the statistical motivation for the evaluation function is discussed in this Section. Section 3 describes a Newton-like method for the optimisation of the evaluation function in parameter space. Section 4 shows how the evaluation function is integrated into a Kalman filter. Section 5 describes the method by which the state variables of the Kalman filter are initialised. Finally, Section 6 briefly reviews the results obtained on the PETS2000 test sequence and the advantages of the method. Full results will be presented as a movie at the workshop.

2. Statistical evaluation function

The method can be used to optimise any set of parameters which determine the image coordinates of the model contours, but for the sake of concreteness we shall discuss the specific application relevant to this workshop: tracking motor vehicles with 3-D facet models.

The pose of an object to be tracked on the ground plane is defined by 2 translational and 1 rotational degrees of freedom, giving 3 pose parameters: the vector of parameters is $\mathbf{p} = (X, Y, \theta_Z)$. These 3 parameters define the space which must be searched for the best match between model and image.

The evaluation of the match is based on the components of the gradient of image grey levels, orthogonal to the model contours projected onto the image plane. Define the normal $\mathbf{v}$ to a projected model line at a sample point on a model contour, the distance ν on this normal from an arbitrary origin and the distance μ of the model contour from the same origin (so that $\nu - \mu$ is the distance from the line along the normal). Given a function ϵ of the grey-level values in the neighbourhood of image position ν and a function w of the distance from the model line, a general form for the evaluation function is

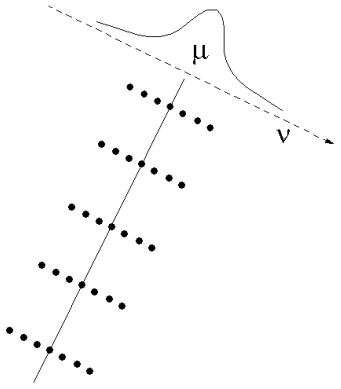

Figure 1. Diagram illustrating the meaning of the symbols μ and ν and the spacing and sampling of the normals to a projected model line: one evaluation is performed on each normal, using the Gaussian window.

$$e = \sum_{\nu} \epsilon(\nu)\, w\left(\frac{\nu - \mu}{\sigma}\right) \tag{1}$$

where σ is a scaling factor (in pixels) and the sum is over a discrete set of samples, separated by $\delta = \sigma/4$ and extending $2\sigma = 8\delta$ on either side of the model contour:

$$\nu \quad \in \quad \{\mu - 8\delta,\, \mu - 7\delta,\, \ldots,\, \mu,\, \mu + \delta,\, \ldots,\, \mu + 8\delta\} \tag{2}$$

Figure 1 illustrates the meaning of the definitions of μ and ν; δ is the distance between the dots on the normals to the model contour. The parameter μ is introduced because this parameterisation makes explicit the dependency of the evaluation function on the position of the model line, even though ν could have been more simply defined as distance from the model contour.

It is desirable, for noise immunity and efficient optimisation, that e be a smooth function of image grey levels and of the image location μ of the contour.

Smoothness with respect to grey levels is determined by $\epsilon(\nu)$. We use the absolute value of the discrete derivative, in the direction $\mathbf{v}$, of the image grey-level values I:

$$\epsilon(\nu) = |\, I(\nu - \delta/2) - I(\nu + \delta/2)\,| \tag{3}$$

In terms of smoothness, the square of the grey-level differences would be preferable to the absolute value. However, the important advantage of the absolute value is that it has a simple statistical interpretation (see Subsection 2.1) and, experimentally, leads to faster and more robust convergence.

Smoothness with respect to μ (the normal component of the image translation of the contour) is determined by the smoothness of the window function. In addition to being smooth, the window function should be even[1], non-negative, integrable and monotonically decreasing away from the origin. We use a Gaussian window:

$$w(a) = \exp\left(\frac{-a^2}{2}\right) \tag{4}$$

In addition to smoothing the evaluation function with respect to μ, the window function also accounts for geometric errors in the model and blurring of the image. The parameter σ can be related to the (expected) distance of the object from the camera ΔY_c by the relation

$$\sigma = \frac{F}{\Delta Y_c}\sigma_{\mathbf{x}} \tag{5}$$

where F is the focal length of the camera, so that σ in image coordinates is equivalent to $\sigma_{\mathbf{x}}$ in world coordinates. Good results have been achieved by gradually reducing the parameter $\sigma_{\mathbf{x}}$ from 0.5 to 0.1 meters.

Other parameters of the evaluation function are the sampling scheme on the model lines (*i.e.* the spacing between the line-points that are evaluated) and the sampling scheme on the normals (needed to compute the discrete grey-level derivatives). As pointed out above, the spacing between samples on the normals was fixed at $\delta = \sigma/4$ and samples were taken up to 2σ from the model contour. The sampling scheme for the model lines is described in Subsection 2.1.

Note that the evaluation function is insensitive to the direction of contrast, and also to the type of image feature, *i.e.* whether the contrast arises from a step or ramp edge, an image line, or any other grey-level profile, as long as the sum of grey-levels differences under the window is constant.

2.1. Combining evaluations over several sampling points

Point evaluations obtained at different positions on a projected model contour must be combined into a single evaluation for the contour, which, ideally, should have a statistical interpretation. The main problem arises in integrating point evaluations over straight line segments, since these evaluations are not statistically independent and their correlation depends on the distances between the sample points.

Let us introduce two variables L and $\bar{e}$: L is the sampled length of the line-segment, *i.e.* the largest distance between two sample points; $\bar{e}$ is the average point evaluation for the line-segment:

$$\bar{e} = \frac{1}{m}\sum_{i=1}^{m} e \tag{6}$$

[1]Note however that an argument for an asymmetric evaluation function has been made in [14]

where m is the number of point-evaluations over the segment.

In natural images, partial correlations between evaluations can be expected to exist at all spatial scales [19]. In this case, it is necessary to estimate the probability distribution of $\bar{e}$ empirically. The relevant probability is the P-value, *i.e.* the probability of obtaining a value of $\bar{e}$ no lower than the observed value. Fortunately, as shown in previous research [16, 17], the distribution can be easily parameterised: log-P-values are linearly related to values of $\bar{e}$ over a large range, and the slope of the linear relationship is inversely proportional to the square root of L. The slope and intercept are image-dependent, but can be assumed to be constant over a short image sequence. This relationship allows an efficient sampling scheme for the model contours: samples on a line segment can be spaced by a distance proportional to the square root of the length of the projected line segment.

Given this linear relationship, it is easy to convert the line evaluation $\bar{e}$ into the log-P-value that a value equal or greater than $\bar{e}$ is obtained by random chance:

$$\log P\,(\,\bar{e}\,) = -\max\left[0,\ \beta_L + \alpha_L\,\bar{e}\right] \tag{7}$$

where α_L, β_L are functions of the image and of the segment length. The negative log-P-value for a model line-segment will be referred to as the *line score*.

As pointed out above, α_L is approximately proportional to the square root of L. The value of β_L as a function of L can be computed from the value of α_L and eqn 7 by keeping in mind that the expected value of $\bar{e}$ is a constant: $\langle\,\bar{e}\,\rangle = \langle\,e\,\rangle$ irrespective of L.

2.2. Combining evaluations over several model lines

In combining line scores, we assume that they are statistically independent. This assumption is questionable for two reasons: first, several model line-segments are close and parallel; second, texture varies from one image region to the other, so that line scores are likely to be correlated if obtained in the same image region. However, the independence assumption works in practice. Under this assumption, the sum of the scores for all line segments

$$E = -\sum_i \log P\,(\,\bar{e}_{\,i}) \tag{8}$$

is the negative log-probability of obtaining the observed scores, or higher scores, for each of the lines; it is *not* the negative log-probability of obtaining the observed sum of scores, or a larger sum. It is easy to see that the probability of obtaining the observed sum (or higher) must be greater than the probability of observing each of the scores (or higher scores), because the observed sum could also be obtained by permutations of the scores.

As long as the number of visible model contours remains the same, the above consideration does not make any difference in practice. However, a problem arises when a change of pose results in a change in number of visible model contours, due to occlusion. An increase in the number of visible model lines always results in an increase of E, unless the scores for the additional visible model lines are zero. In other words, a correction is needed to take into account the number of visible model lines.

Dividing E by the number of visible model lines would not result in a proper probabilistic measure, but fortunately it is possible to combine line scores with a technique introduced by Fisher ([5], *pp.*99-100) and based on two principles:

- negative log-P-values have an exponential distribution, hence they are distributed as $-\chi^2/2$ with 2 degrees of freedom;
- the sum of χ^2-distributed independent variables is itself a χ^2-distributed variable, with a number of degrees of freedom which is the sum of the degrees of freedom of the summed variables.

Therefore, the variable $-2E$ (where the sum is over all point evaluations) has a χ^2 distribution with $2n$ degrees of freedom; its negative log-P-value of E can be computed by using the repeated-fraction approximation to the incomplete gamma function ([18], *pp.*213-222):

$$\begin{aligned} G &= -\log P(E) \\ &= -\log \frac{\Gamma(n,E)}{\Gamma(n)} \end{aligned} \tag{9}$$

where G is the negative log-P-value of E, which we define as the model Evaluation: the capital letter emphasises that it is the evaluation of the model, rather than the evaluation of a line-segment (line score) or a point on a line segment.

Since the gamma function is a monotonic function of its second argument, it follows that E and G have the same maxima if n is constant.

2.3. Converting the Evaluation into a likelihood

We have seen that G is the negative log-probability of obtaining at least the observed sum of scores by random chance, but what is of interested is the log-probability that the object being tracked is at the pose where G has been measured. In this Subsection, the relationship between G and the log-likelihood of the hypothesis pose will be derived.

First, the "data" must be defined explicitly. In principle, the data $\mathbb{D}$ is the set of all the Evaluations that could be computed at any point in parameter space (thus covering the whole image). When estimating the probability of the data, it is necessary to keep in mind that these Evaluations, are not independent, due to the value of σ: Evaluations which are observed at "close" points in parameter space arise from overlapping grey-level measurements. Therefore, we consider only Evaluations observed on a grid $\mathbb{P}$ in parameter space, each element $\mathbf{p} \in \mathbb{P}$ being sufficiently far from its neighbours for the corresponding Evaluations to arise from independent measurements. As we shall see below, the precise way in which $\mathbb{P}$ is defined is immaterial, as long as one element of the grid is the pose that maximises G: only this piece of data appears in the final equation for the likelihood of the hypothesis pose.

Given a hypothesis $\mathbf{p}_h \in \mathbb{P}$ about the object pose, the prior and posterior probabilities of the data are the products of the probabilities of obtaining all the Evaluations on the grid:

$$P(\mathbb{D}) = \prod_{\mathbf{p}\in\mathbb{P}} P\left[G(\mathbf{p})\right] \tag{10}$$

$$P(\mathbb{D}\,|\,\mathbf{p}_h) = \prod_{\mathbf{p}\in\mathbb{P}} P\left[G(\mathbf{p})\,|\,\mathbf{p}_h\right] \tag{11}$$

Note that all the poses $\mathbf{p} \in \mathbb{P}$ are used to define the probabilities, even though only one pose $\mathbf{p}_h$ is assumed, by hypothesis, to be the true pose.

As shown in Subsection 2.2, the log-prior probability of the data is given by

$$\log P(\mathbb{D}) = -\sum_{\mathbf{p}\in\mathbb{P}} G(\mathbf{p}) \tag{12}$$

where the sum is over all elements of the grid $\mathbb{P}$.

The posterior probabilities of the Evaluations are the same as the prior probabilities everywhere on the grid, except at the hypothesis pose: if the object is located at the hypothesis pose, the probability of observing a given Evaluation at any other location on the grid is not affected. Therefore

$$\log P(\mathbb{D}\,|\,\mathbf{p}_h) = \log P\left[G(\mathbf{p}_h)\,|\,\mathbf{p}_h\right] - \sum_{\mathbf{p}\neq\mathbf{p}_h} G(\mathbf{p}) \tag{13}$$

In the absence of information about the colour of the object, the lighting conditions, unmodelled occlusion, *etc.*, any Evaluation is assigned equal P-value P_h at the hypothesis pose:

$$P_h \equiv P\left[G(\mathbf{p}_h)\,|\,\mathbf{p}_h\right] \tag{14}$$

Combining eqns 9, 12, 14 and 13, we obtain

$$\log P(\mathbb{D}\,|\,\mathbf{p}_h) = G(\mathbf{p}_h) + \log P_h + \log P(\mathbb{D}) \tag{15}$$

In words, the likelihood of the hypothesis pose, given the data, is equal to the model Evaluation at the hypothesis

pose, except for an additive constant given by the second and third terms on the right-hand side of eqn 15.

In conclusion, if we accept eqn 14 for the Evaluation at the hypothesis pose, then the pose that maximises G is the maximum-likelihood estimate of the model pose.

3. Optimisation by a Newton-like method

Optimisation of G can be carried out efficiently by the use of a Newton-like method [16, 17]. The assumption behind Newton's method is that the Evaluation function can be approximated, close to a local maximum $\mathbf{p}_0$, by two terms of its Taylor expansion:

$$G(\mathbf{p}_h) \approx G_0 + \Delta\mathbf{p}^T \cdot \frac{1}{2}\mathbf{H} \cdot \Delta\mathbf{p} \tag{16}$$

where $G_0 = G(\mathbf{p}_0)$, $\Delta\mathbf{p} = \mathbf{p}_h - \mathbf{p}_0$ and $\mathbf{H}$ is a square symmetrical matrix describing a parabolic approximation to the Evaluation function in parameter space.

From eqn 16 it follows by differentiation that the gradient at the current pose is:

$$\nabla_{\mathbf{p}} G \approx \mathbf{H} \cdot \Delta\mathbf{p} \tag{17}$$

If the gradient and curvature of G can be estimated, eqn 17 gives a system of linear equations that can be solved for $\Delta\mathbf{p}$ to estimate the offset between the current hypothesis pose and the optimal pose $\mathbf{p}_0$.

The curvature is usually computed as the Hessian of G. To see why this is not sensible in our case, let us derive the gradient and Hessian of the Evaluation function.

3.1. Gradient of model Evaluation

The gradient of G in pose space is equal to the gradient in the space of line-point positions $\underline{\mu}$, projected onto pose space by means of the Jacobian $\mathbf{J}$ (the matrix of first derivatives of $\underline{\mu}$ with respect to $\mathbf{p}_h$) and scaled by the derivative $\dot{G}$ of G with respect to E:

$$\nabla_{\mathbf{p}} G = -\dot{G} \cdot \mathbf{J}^T \cdot \nabla_{\underline{\mu}} E \tag{18}$$

where

$$\dot{G} = -\frac{\partial}{\partial E} \log \frac{\Gamma(n, E)}{\Gamma(n)} \tag{19}$$

The Jacobian is a linear approximation to inverse perspective and is derived in [17]. It should be noted that, at each iteration, full perspective is used to update the projection of the model lines. Therefore, the method will converge to the parameters giving the highest Evaluation function under full perspective.

The elements of the gradient of E in $\underline{\mu}$-space are the first derivatives of point evaluations with respect to line-point positions on the normal, normalised by the slope of the piecewise-linear relationship, eqn 7; a different slope applies to each line segment, depending on its length.

3.2. Hessian of model Evaluation

Using the rule for the derivative of a product of functions, it is easy to derive the Hessian of G in pose space:

$$\begin{aligned} \nabla^2_{\mathbf{p}} G = \quad & \nabla_{\mathbf{p}} \dot{G} \cdot \mathbf{J}^T \cdot \nabla_{\underline{\mu}} E \\ + \quad & \dot{G} \cdot \nabla_{\mathbf{p}} \mathbf{J}^T \cdot \nabla_{\underline{\mu}} E \\ + \quad & \dot{G} \cdot \mathbf{J}^T \cdot \nabla_{\mathbf{p}} \nabla_{\underline{\mu}} E \end{aligned} \tag{20}$$

Eqn 20 can be well approximated by eliminating the first and second terms on the right-hand side (see also [18], *pp.*682-683):

$$\begin{aligned} \nabla^2_{\mathbf{p}} G \quad &\approx \quad \dot{G} \cdot \mathbf{J}^T \cdot \nabla_{\mathbf{p}} \nabla_{\underline{\mu}} E \\ &\approx \quad \dot{G} \cdot \mathbf{J}^T \cdot \nabla^2_{\underline{\mu}} E \cdot \mathbf{J} \end{aligned} \tag{21}$$

where $\nabla^2_{\underline{\mu}} E$ is an $M \times M$ diagonal matrix whose diagonal elements are the second derivatives of point evaluations with respect to line-point positions on the normal, again normalised by the slope of the piecewise-linear relationship (eqn 7), as in the case of the gradient.

3.3. Approximating the Hessian

In the practical application of the method, a problem arises from the diagonal elements of $\nabla^2_{\underline{\mu}} E$, *i.e.* the second derivatives of e with respect to μ:

$$\tilde{\alpha}\, \frac{\partial^2 e}{\partial \mu^2} = \tilde{\alpha}\, \frac{1}{\sigma^2} \sum_{\nu} \epsilon(\nu)\, \ddot{w}\left(\frac{\nu - \mu}{\sigma}\right) \tag{22}$$

where $\tilde{\alpha}$ denotes the first derivative of the line score with respect to $\bar{e}$ (see eqn 7), which is either equal to α or to zero, and $\ddot{w}$ is the second derivative of w with respect to μ [16, 17]. Like the expression for e, this is a weighted sum of grey-level differences, but the weights in eqn 22 can be positive, negative or zero. The truncated Taylor series is a good approximation only if most of the significant image derivatives are weighted with negative weights over all model line-points, since Newton's method can only be used for maximisation if the Hessian is negative definite. In fact, even if most of the image derivatives are negatively weighted, the remaining, positively weighted derivatives will contribute to the Hessian, making its eigen-values smaller (in absolute value) and therefore making Newton's method unstable. In other words, truncating the Taylor expansion does not give a good approximation even close to, or at, a stable point.

As shown in previous work [16, 17], this problem can be avoided by a simple strategy. Define a centre of mass ν_0

$$\nu_0 = \frac{\sum \nu \cdot k(\nu)}{\sum k(\nu)} \tag{23}$$

where $k(\nu)$ is defined with reference to the original point evaluation function (eqn 1):

$$k(\nu) = \epsilon(\nu)\, w(\nu - \mu) \tag{24}$$

It is easy to minimise the weighted sum of squared distances between points on the model lines and centres of mass on the normals:

$$\tilde{G} = \dot{G} \ \sum K \cdot (\nu_0 - \mu)^2 \tag{25}$$

with the weight for each line-point being

$$\begin{aligned} K &= \tilde{\alpha} \sum k(\nu) \\ &= \tilde{\alpha}\, e \end{aligned} \tag{26}$$

Minimisation of $\tilde{G}$ is a linear least-squares problem and therefore a single iteration should be sufficient (except for the error introduced by the small-angle approximation when the gradient and Hessian are projected onto parameter space). However, once an iteration has taken place, the model contours have moved in the image plane and therefore there will be a new, different centre of mass, as well as a different weight K, for each point on the model contours. The sum of weighted squared distances from the new centres of mass can be minimised again, until the algorithm converges. Since the gradient of $\tilde{G}$ is the same as the gradient of the original, statistically-motivated objective function G, the stable points of this algorithm are the same as the stable points of G.

We define the Hessian of $\tilde{G}$ in parameter space as $\tilde{\mathbf{H}}$. The method as implemented can be described as a Newton-like (or approximate Newton) method [9].

4. The Kalman filter

Although there is as yet no theoretical justification for this observation, we have observed empirically that $G(\mathbf{p})$ can be locally approximated in parameter space by a parabola with curvature $\mathbf{H} = \tilde{\mathbf{H}}_0$, where $\tilde{\mathbf{H}}_0$ is the Hessian of $\tilde{G}$ estimated at convergence, *i.e.* at $\mathbf{p}_0$. Therefore, we use $\tilde{\mathbf{H}}_0$ as an estimate of the inverse covariance of the likelihood:

$$\log P(\mathbb{D} \,|\, \mathbf{p}_h) \ \approx \ \log P(\mathbb{D} \,|\, \mathbf{p}_0) + \Delta\mathbf{p}^T \cdot \frac{1}{2}\tilde{\mathbf{H}}_0 \cdot \Delta\mathbf{p} \tag{27}$$

where, from eqn 15

$$\log P(\mathbb{D} \,|\, \mathbf{p}_0) = G_0 + \log P_h + \log P(\mathbb{D})$$

The complete state vector for the object includes pose parameter and velocities: $\mathbf{s} = (X, Y, \theta_Z, v, \omega, a)$, where v is the tangential velocity, ω the angular velocity and a the tangential acceleration. A hypothesis state is related to a hypothesis pose by the observation matrix $\mathbf{B}$:

$$\mathbf{p}_h = \mathbf{B}^T \cdot \mathbf{s} \tag{28}$$

where $\mathbf{B}^T$ is a 3×6 matrix which turns the state vector $\mathbf{s}$ into the pose vector $\mathbf{p}_h$ by eliminating the last 3 state variables.

The p.d.f. (probability density function) of the object state must be approximated by a Gaussian distribution for the application of the Kalman filter. In the following, the p.d.f. of the state is denoted by $f_\ominus$ after the prediction update and by $f_\oplus$ after the innovation update. We further define further the means $\widehat{\mathbf{s}}_\ominus$ and $\widehat{\mathbf{s}}_\oplus$ the covariances $\mathbf{\Sigma}_\ominus$ and $\mathbf{\Sigma}_\oplus$, and the normalisation factors $Z_\ominus$ and $Z_\oplus$, for $f_\ominus$ and $f_\oplus$, respectively: the log-p.d.f. of the state after the prediction update is given by

$$\log f_\ominus \ = \ -\log Z_\ominus - (\mathbf{s} - \widehat{\mathbf{s}}_\ominus)^T \cdot \frac{1}{2}\mathbf{\Sigma}_\ominus^{-1} \cdot (\mathbf{s} - \widehat{\mathbf{s}}_\ominus)$$

and the p.d.f. of the state after the innovation update is given by

$$\log f_\oplus \ = \ -\log Z_\oplus - (\mathbf{s} - \widehat{\mathbf{s}}_\oplus)^T \cdot \frac{1}{2}\mathbf{\Sigma}_\oplus^{-1} \cdot (\mathbf{s} - \widehat{\mathbf{s}}_\oplus)$$

The prediction step of the Kalman filter requires a state-transition matrix and a covariance matrix for the system input. Both can be obtained by making reasonable assumptions about the underlying dynamics of the motor vehicle. The state-transition matrix is given by:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta t \cos\theta_Z & \Delta t v \sin\theta_Z & 0 & 0 \\ 0 & 1 & \Delta t \sin\theta_Z & -\Delta t v \cos\theta_Z & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & \xi & 0 \\ 0 & 0 & 0 & 0 & 0 & \xi \end{pmatrix}$$

where Δt is the time interval between video frames and

$$\xi = \exp(-\Delta t / \tau)$$

with τ as the time constant of decay of the state variables with zero means (angular velocity and tangential acceleration). The diagonal elements are unity for state variables that have non-zero means (pose variables and tangential velocity). Note that $\mathbf{F}$ is not constant but depends on the state variable θ_Z, and therefore on time: this is the constraint that the car cannot move sideways.

In the case of a motor vehicle, the inputs can be simplified as tangential accelerations and angular velocity, as determined by the driver. Therefore, all elements of the (6×6) input covariance matrix $\mathbf{Q}$ are zero, except for two diagonal elements: $\mathbf{Q}_{6,6} = \sigma_a^2$ for the tangential acceleration

and $\mathbf{Q}_{5,5} = \sigma_\omega^2$ for the angular velocity. In practice, better performance has been obtained by setting $\mathbf{Q}_{3,3}$ to a small value, so that the innovation about the vehicle orientation is always weighted more heavily than the prediction.

The matrices $\mathbf{B}$, $\mathbf{F}$ and $\tilde{\mathbf{H}}_0$ can be inserted into the well-known Kalman-filter equations for the prediction update:

$$\begin{aligned} \mathbf{\Sigma}_\ominus(t+\Delta t) &= \mathbf{F}(t)\cdot\mathbf{\Sigma}_\oplus(t)\cdot\mathbf{F}^T(t)+\mathbf{Q} \quad (29)\\ \widehat{\mathbf{s}}_\ominus(t+\Delta t) &= \mathbf{F}(t)\cdot\widehat{\mathbf{s}}_\oplus(t) \quad (30)\end{aligned}$$

and for the innovation update (with the time arguments omitted for simplicity):

$$\begin{aligned} \mathbf{W} &= \left(\mathbf{B}^T\cdot\mathbf{\Sigma}_\ominus\cdot\mathbf{B}+\tilde{\mathbf{H}}_0^{-1}\right)^{-1}\\ \mathbf{\Sigma}_\oplus &= \mathbf{\Sigma}_\ominus-\mathbf{\Sigma}_\ominus\cdot\mathbf{B}\cdot\mathbf{W}\cdot\mathbf{B}^T\cdot\mathbf{\Sigma}_\ominus \quad (31)\\ \widehat{\mathbf{s}}_\oplus &= \widehat{\mathbf{s}}_\ominus+\mathbf{\Sigma}_\ominus\cdot\mathbf{B}\cdot\mathbf{W}\cdot(\mathbf{p}_0-\mathbf{B}^T\cdot\widehat{\mathbf{s}}_\ominus) \quad (32)\end{aligned}$$

5. Initialisation

The tracker is initialised when a new vehicle is detected and its pose and velocity are estimated. The detection and initial estimation are accomplished by subtracting a reference image from the current image: a new object is detected when the image difference has a significant amount of energy, clustered at a location distinct from the locations of all currently tracked objects. When the energy cluster is no longer in contact with the image boundaries, the centre of mass of this energy is computed (in image coordinates) and the position of the object is assumed to be on the ray defined by the centre of mass, at an elevation of 0.2 meters with respect to the ground plane. The orientation of the object and its velocity are estimated by tracking its position, determined as above, for three frames. Then the pose is optimised, under the assumption that the object is a vehicle, as opposed to a pedestrian or a false alarm; this assumption is corroborated or falsified by the value of the Evaluation function at convergence.

The difference image is also used to re-initialise the object pose when the Kalman prediction fails to provide a good initialisation for pose refinement: this happens very seldom in the PETS2000 sequence.

6. Results and conclusions

The vehicles in the test sequence are tracked at 5 frames/second (*i.e.* with innovation every 5th frame of the test sequence) with occasional misalignments, from which the system always recovered, except when the vehicles are very far from the camera. The Evaluation function allows discrimination between motor vehicles and pedestrians, but in the test sequence, this is accomplished more simply by estimating the total energy in the relevant region of the difference image. In fact, image differencing is remarkably successful in determining the initial pose, orientation and velocity of the vehicles in the PETS2000 test sequence. However, these initial estimates involve somewhat arbitrary constant parameters, for instance the estimated elevation of the centre of mass in the difference image (see previous Subsection). Typical results are shown in Figures 2 and 3.

The main advantages of the contour-based tracker are the statistical interpretation of the goodness-of-fit measure, its operation in 3-D and the single, simple concept on which the method is based: because of the top-down mode of operation, the only important assumption is that image edges are likely to be detectable at internal or external object boundaries; other assumptions, such as the constant P-value for the Evaluation at the hypothesis pose, could be revised without changing the essence of the method.

From the statistical point of view, one attractive feature of the method is that it obviates the need for marginalisation. Methods that involve feature detection at any stage must marginalise over all possible correspondences of image features to model features, compatible with a hypothesis pose, in order to compute the correct likelihood of the pose. Similarly, 2-D contour methods must marginalise over all parameters of the 2-D contours, compatible with a state of a 3-D object, in order to compute the correct likelihood of the state. Ultimately, a tracking system should make inferences about the real 3-D world: the capability to relate directly a 3-D model to image grey-levels is a desirable feature.

Using 3-D models has several other advantages:

- in the case of rigid objects such as motor vehicles, the optimisation of the model parameters actually becomes simpler, because only the pose parameters change from frame to frame: even when shape parameters need to be estimated, the estimation is a more constrained problem if the shape of an object is known to remain constant over the entire image sequence;
- reasoning about occlusion and collisions between objects is much simpler in 3-D;
- in 3-D, the Kalman filter incorporates a physical model of the object being tracked.

It is interesting that the equations for the classical Kalman filter are obtained by assuming a constant P-value for the model Evaluation obtained at the hypothesis pose. At first sight, it would seem that, paradoxically, the posterior probability of the data is model-independent, while the prior probability of the data is model-dependent. In fact, this is not the case if the data is defined properly, *i.e.* as the set of all the model Evaluations that could be made: this remains true even though most of the possible Evaluations are

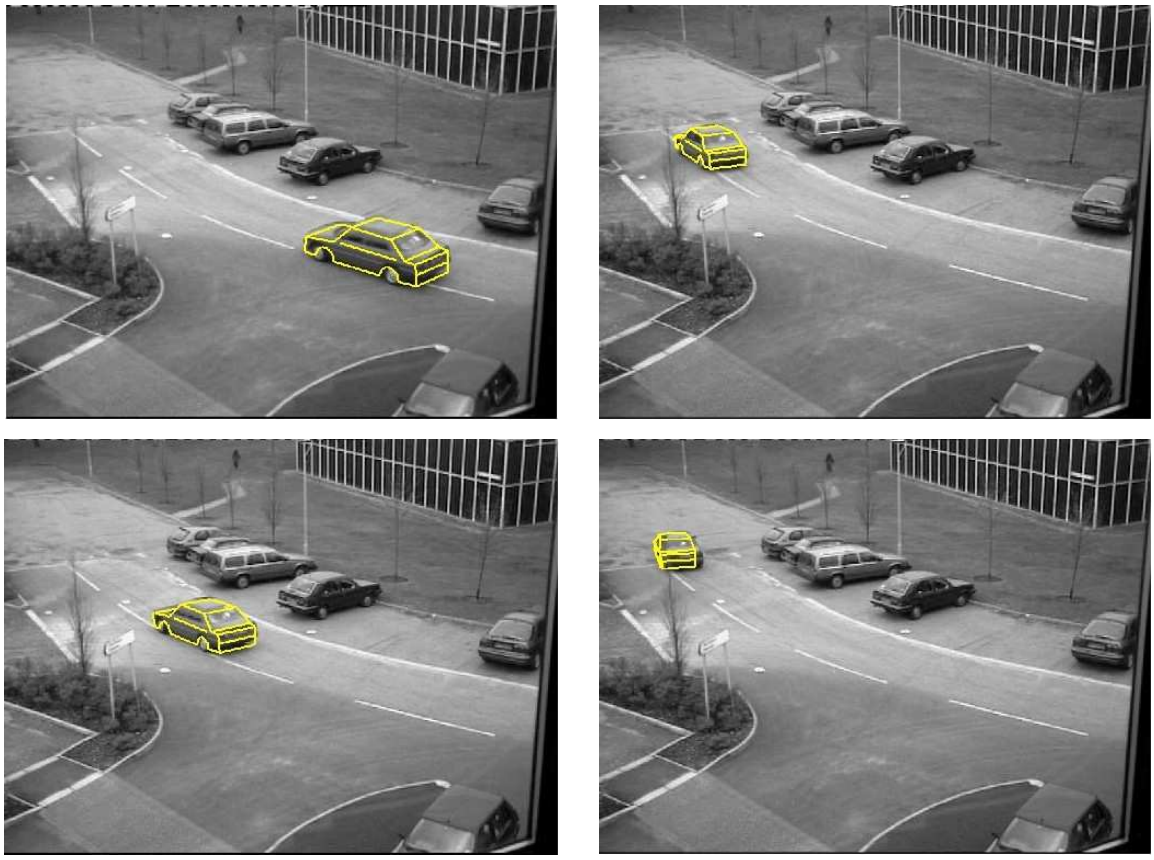

Figure 2. Tracking of the saloon (sedan) in the PETS2000 test sequence. The figure shows every 25th frame.

not carried out, because they do not actually carry information about the hypothesis. In other words, the data remain the same as long as the image remains the same; what is model-dependent is only the piece of data useful to evaluate the hypothesis. This concept, developed in Subsection 2.3, shows that objections to methods with a model-dependent prior probability [22] do not apply to our method.

The tracking system is still at an early stage of development; in particular, the method used for initialisation should be improved. Other desirable improvements include the use of colour images as well as modelling object shadows and incompenetrability between objects. To increase robustness, it would also be desirable to integrate evidence provided by different algorithms: an advantage of the Bayesian framework is that it allows efficient combination of statistical evidence from different sources.

References

[1] Y. Bar-Shalom, T. E. Fortmann, *Tracking And Data Association.* Academic Press, 1988.

[2] A Baumberg, DC Hogg , Learning Flexible Models from Image Sequences, Proc. of the European Conference on Computer Vision (ECCV), 1994.

[3] A Blake, M Isard, *Active Contours*, Springer-Verlag, Berlin 1998.

[4] KS Brisdon, Hypothesis verification using iconic matching. Ph.D. thesis, University of Reading, 1990.

[5] R Fisher, *Statistical methods for research workers* (14th edition). Macmillan, New York, 1970.

[6] WEL Grimson, The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments, IEEE Trans. PAMI 13(9): 920-935, 1991.

[7] C Harris, Tracking with rigid models. In: *Active Vision* (A Blake, A Yuille, eds) pp 59-73. MIT Press, 1992.

[8] M Isard, A Blake, CONDENSATION – conditional density propagation for visual tracking. Int. J. Computer Vision, 29(1): 5-28, 1998.

[9] M Kawato, T Inui, S Hongo, H Hayakawa, Computational theory and neural network models of interaction between visual cortical areas. ATR Technical Report TR-A-0105, ATR Laboratories, Soraku-gun, Kyoto 619-02, Japan, 1991.

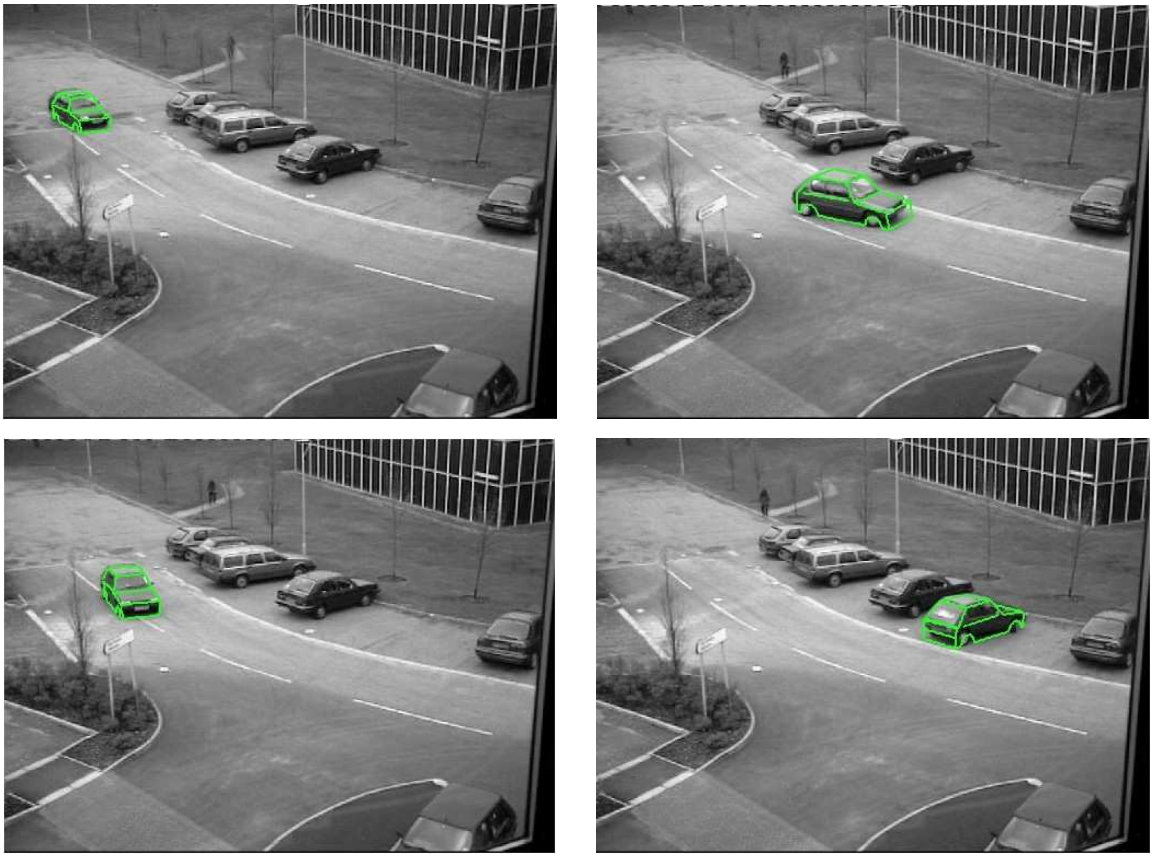

Figure 3. Tracking of the hatchback (compact) in the PETS2000 test sequence. The figure shows every 75th frame.

[10] D Koller, K Daniilidis, H-H Nagel, Model-based object tracking in monocular image sequences of road traffic scenes. Int.J.Comp.Vis. 10(3): 257-281, 1993.

[11] H Kollnig, H-H Nagel, 3D pose estimation by fitting image gradients directly to polyhedral models. Proc. 5th Int. Conf. Comp. Vision (ICCV), pp 569-574, IEEE Computer Soc. Press, 1995.

[12] DG Lowe, *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[13] DG Lowe, Fitting parametrized 3-D models to images. IEEE Trans. PAMI 13(5): 441-450, 1991.

[14] J. MacCormick. A. Blake, A probabilistic contour discriminant for object localisation. Proc 6th Int. Conf. Computer Vision (ICCV), 1998.

[15] AEC Pece, GD Sullivan, Model-based control of an active camera head. Proc. EU-HCM SMART workshop, Lisbon, April 1995.

[16] AEC Pece, AD Worrall, A statistically-based Newton method for pose refinement. Image and Vision Computing 16: 541-544, 1998.

[17] AEC Pece, AD Worrall, A Newton method for pose refinement of 3D models. Proc 6th International Symposium on Intelligent Robotic Systems, 21-23 July 1998.

[18] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C* (2nd edition). Cambridge University Press, 1992.

[19] DL Ruderman, The statistics of natural images. Network 5: 517-548, 1994.

[20] RS Stephens, Real-time 3D object tracking. Proc. Alvey Vis.Conf. pp 85-90, 1989.

[21] GD Sullivan, Visual interpretation of known objects in constrained scenes. Phil.Trans.R.Soc.Lond. B 337:361-370, 1992.

[22] J Sullivan, A Blake, M Isard, J MacCormick, Object Localization by Bayesian Correlation. Proc Int. Conf. Computer Vision (ICCV), 1999.

[23] AD Worrall, JM Ferryman, GD Sullivan, KD Baker, Pose and structure recovery using active models. Proc. 6th British Machine Vision Conf. (BMVC), pp 137-146, 1995.