Tracking with the EM Contour Algorithm

Arthur E.C. Pece[1] and Anthony D. Worrall[2]

[1] Institute of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen, Denmark
aecp@diku.dk
[2] Department of Computer Science
University of Reading
P.O. Box 225
Reading RG6 6AY, England
anthony.worrall@reading.ac.uk

Abstract. A novel active-contour method is presented and applied to pose refinement and tracking. The main innovation is that no "features" are detected at any stage: contours are simply assumed to remove statistical dependencies between pixels on opposite sides of the contour. This assumption, together with a simple model of shape variability of the geometric models, leads to the application of an EM method for maximizing the likelihood of pose parameters. In addition, a dynamical model of the system leads to the application of a Kalman filter. The method is demonstrated by tracking motor vehicles with 3-D models.

1 Introduction

Active contour methods can be divided into two main classes, depending on the principle used for evaluating the image evidence. One class (*e.g.* [3,2,7,11,21, 20]) relies on the extraction of features from the image in the neighborhood of the contours and the assignment of one, and only one, correspondence between points on the contours and image features. Apart from the problem of finding correct correspondences, the thresholds necessary for feature detection inevitably make these methods sensitive to noise. Furthermore, a correct estimate of the model likelihood requires marginalization over possible correspondences *and* over possible features.

Other active-contour methods (*e.g.* [6,10,12,15,22]) avoid feature detection by maximizing feature *values* (without thresholding) underlying the contour, rather than minimizing the distance between locally-strongest feature and contour. Some form of smoothing (of the image or of the feature field) is necessary to ensure that the objective function is a smooth function of pose parameters. One disadvantage of these methods is that optimization of the contour positions is usually not gradient-based, and therefore slow to converge.

The EM contour algorithm introduced in this paper belongs to this latter class, with the important difference that smoothing is replaced by marginalization over possible deformations of the object shape. This model assumption,

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 3–17, 2002.

apart from being realistic, leads to an objective function which can be interpreted as a squared-distance measure, as in the case of contour methods of the first class.

The EM contour algorithm is a development of previous work [15,16,17]. The main development consists in the formulation of a generative model, leading to a sound theoretical basis and minor modifications of the mathematics.

The two basic model assumptions behind the method are simply stated:

1. Grey-level values of nearby pixels are correlated if both pixels belong to the object being tracked, or both belong to the background, but there are no correlations if the two pixels are on opposite sides of the contour. The form of correlations between nearby pixels is well known from research on the statistics of natural images (*e.g.* [8,9]); therefore, the log-likelihood of the pose and shape parameters of the object being tracked is easily obtained from the image.
2. The shape of the contour is subject to random local variability. Marginalizing over local shape variations (deformations) of the contours leads to the application of the EM method [4,13]. The covariance of the estimate of pose parameters is easily obtained and leads to proper weighting of the innovation in a Kalman filter.

The contour model introduced in this paper has a potentially wide range of applications in machine vision. This paper shows an application to tracking of rigid objects bound to the ground plane.

1.1 Organization of the Paper

Section 2 describes the generative model underlying the tracker and derives the log-likelihood for the state variables of the model. Section 3 introduces the EM contour algorithm used to optimize the log-likelihood in state space. Section 4 describes the dynamical model which underlies the Kalman filter, and lists the parameters of the tracker. Section 5 describes the method by which the state variables are initialized. Finally, Section 6 describes the results obtained on the PETS2000 test sequence and reviews the theoretical advantages of the method.

2 Generative Model

The tracker is based on finding the *MAP* (maximum *a posteriori*) estimate of the state parameters (*i.e.* pose, velocity, acceleration) of the object being tracked, given the image sequence up to the current frame. By Bayes' theorem, the posterior pdf (probability density function) is the product of the prior pdf and the likelihood (divided by a constant which can be neglected for optimization purposes).

In order to derive expressions for the prior pdf and likelihood of the state parameters, a generative model is introduced, consisting of two main components:

- a dynamical model, which defines the pdf over states of the object being tracked at a given point in time, given the state at an earlier time, *i.e.* the prior pdf;
- an observation model, which defines the pdf over images, given the current state of the object, *i.e.* the likelihood.

The dynamical model is fairly standard and is analyzed in section 4. At this stage, only 3 pose parameters of the object being tracked need to be considered: the X and Y coordinates on the ground plane and the orientation θ_Z. The vector of these pose parameters is defined as $\mathbf{y} = (X, Y, \theta_Z)$.

2.1 Observation Model

The observation model can itself be broken down into two components:

- a geometric component which defines a pdf over image locations of contours, given (a) the pose parameters of the object, (b) a 3-D geometric model of the object including an estimate of its shape variability, and (c) the camera geometry;
- a "coloring/shading" component which defines a pdf over grey-level differences between pixels, given the image locations of the contours.

The average geometric model is the shape of an "average" automobile. Given the current pose parameters of the object and the camera parameters, it is straightforward to project the 3-D boundaries between the facets of the geometric model onto the image plane (with hidden line removal), to obtain the expected image contours.

We begin by describing the case of no shape variability; this simplification will be removed in subsection 2.3.

2.2 Statistics of Grey-Level Differences

Grey levels at two close locations on the image plane are correlated, except for the case of two pixels which are known to lie on two different sides of an object boundary, in which case no correlation is expected. Research on the statistics of natural images shows that the pdf f_L of grey-level differences (gld's) between adjacent pixels is well approximated by a generalized laplacian [8]:

$$f_L(\Delta\mathcal{I}) = \frac{1}{Z_L} \exp\left(-\left|\frac{\Delta\mathcal{I}}{\lambda}\right|^{\beta}\right) \tag{1}$$

where $\Delta\mathcal{I}$ is the gld, λ is a parameter that depends on the distance between the two sampled image locations, β is a parameter approximately equal to 0.5, and Z_L is a normalization constant. For $\beta = 0.5$, it can be easily obtained that $Z_L = 4\lambda$.

Note that the generalized-laplacian pdf is obtained (reliably in all image ensembles which have been investigated) without manual intervention, *i.e.* without

determining which gld's are measured across object boundaries and which gld's are measured within the same object. This means that the pdf includes the effect of *unpredicted* object boundaries, indeed it can be derived from a simple model of random occlusion [9]. This point is relevant in the following.

The pdf given by Eq.1 has been shown to fit the log-ratios of grey levels, rather than grey-level differences, but, for small values of the gld, the logarithmic transformation has only a small effect on the shape of the pdf. In addition, the gamma correction of most cameras already approximates a logarithmic transformation to some extent.

It will be clear at this point that the method is based on placing the contours in the image plane so as to minimize the sum of the log-prior pdf's of the gld's measured across the contours, under the constraints given by the shape model of the object being tracked. Similar principles were employed elsewhere [19,22].

We define the image coordinate system $\mathbf{u}$ and a binary indicator variable $\eta_{\Delta\mathbf{u}}(\mathbf{u})$, with value 1 if the boundary of the object being tracked is located between $\mathbf{u} - \Delta\mathbf{u}/2$ and $\mathbf{u} + \Delta\mathbf{u}/2$, and value 0 otherwise.

Given no boundary of the object of interest between two image locations, the pdf of gld's is given by Eq.1:

$$f[\Delta\mathcal{I}(\mathbf{u}) \,|\, \eta_{\Delta\mathbf{u}}(\mathbf{u}) = 0] = f_L[\Delta\mathcal{I}(\mathbf{u})] \tag{2}$$

where $\Delta\mathcal{I}(\mathbf{u}) = I(\mathbf{u} + \Delta\mathbf{u}/2) - I(\mathbf{u} - \Delta\mathbf{u}/2)$. Note that we are not assuming the total absence of boundaries between the two locations: only the absence of the boundary of the object of interest.

Grey levels observed on opposite sides of an object boundary are statistically independent, and therefore the conditional pdf of gld's, given an object boundary between the two image locations, can be assumed to be uniform[1]:

$$f[\Delta\mathcal{I}(\mathbf{u}) \,|\, \eta_{\Delta\mathbf{u}}(\mathbf{u}) = 1] \approx 1/m \tag{3}$$

where m is the number of grey levels.

Let us consider the normal to a point on a projected line segment, as illustrated in Fig. 1. The coordinate along this normal is denoted by ν, and the point of intersection of the line segment is denoted by μ, so that $\nu - \mu$ is the distance from the line segment along the normal. Given regularly-spaced samples of grey levels on the normal, with spacing $\Delta\nu$ and bilinear interpolation, we define the *observation* (at a given sample point on a contour) as $\boldsymbol{\Delta\mathcal{I}} = \{\Delta\mathcal{I}(i\Delta\nu) \,|\, i \in \mathbb{Z}\}$. Assuming statistical independence between gld's, the pdf F_L of the observation in the absence of a contour is given by:

$$F_L(\boldsymbol{\Delta\mathcal{I}}) \stackrel{\text{def}}{=} \prod_i f_L[\Delta\mathcal{I}(i\Delta\nu)] \tag{4}$$

while the pdf of the observation, given the contour location μ, is given by:

$$\begin{aligned} f(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) &= f[\boldsymbol{\Delta\mathcal{I}} \,|\, \eta_{\Delta\nu}(\mu) = 1] \\ &= \frac{1}{m} F_L(\boldsymbol{\Delta\mathcal{I}}) \; f_L^{-1}[\Delta\mathcal{I}(\mu)] \end{aligned} \tag{5}$$

[1] We ignore a possible dependency on the average grey level of the two image locations.

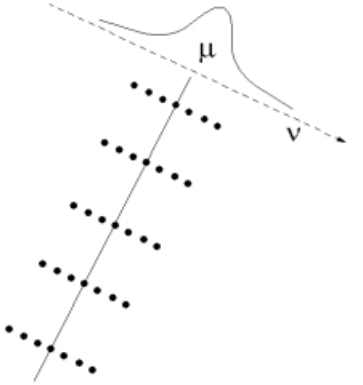

Fig. 1. Diagram illustrating the meaning of the symbols μ and ν and the spacing and sampling of the normals to a contour: one evaluation is performed on each normal, using the Gaussian window.

2.3 Marginalizing over Deformations

The geometric object model cannot be assumed to be perfect. In addition, it is desirable to make the log-likelihood smooth in parameter space. For these reasons, we postulate that the model is subject to random iid (independent, identically distributed) Gaussian deformations at each sample point on the contours: the actual image location of a point on a contour is assumed to have a Gaussian pdf with mean equal to the location predicted from the deterministic geometry, and constant variance. Random shape variability does not preclude additional parameterized shape variations, but shape recovery is not of interest in this paper.

For simplicity, the shape variability is defined in the image plane. For convenience, we define $\epsilon \stackrel{\text{def}}{=} \nu - \mu$, We also define the prior pdf of deformations $f_D(\epsilon)$:

$$f_D(\epsilon) = \frac{1}{Z_D} \exp \frac{-\epsilon^2}{2\sigma^2} \tag{6}$$

where $Z_D = \sqrt{2\pi}\sigma$ is a normalization factor.

The marginalized pdf of an observation is:

$$\begin{aligned} f_M(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) &= \int f(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu + \epsilon)\; f_D(\epsilon)\; d\epsilon \\ &= \frac{1}{m} F_L(\boldsymbol{\Delta\mathcal{I}}) \int f_L^{-1}[\Delta\mathcal{I}(\mu + \epsilon)]\; f_D(\epsilon)\; d\epsilon \end{aligned} \tag{7}$$

The integral can be approximated as a finite sum over the discrete set of possible deformations $\epsilon_j = j\Delta\nu - \mu$ (with $j \in \mathbb{Z}$):

$$f_M(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) = \frac{1}{m} F_L(\boldsymbol{\Delta\mathcal{I}}) \sum_j f_L^{-1}[\Delta\mathcal{I}(j\Delta\nu)]\; f_D(\epsilon_j)\; \Delta\nu \tag{8}$$

The ratio between the marginalized pdf (Eq.8) and the pdf in the absence of a contour (Eq.4) is the likelihood ratio of the hypotheses that there is a contour at location μ and that there is no contour. The likelihood ratio is given by

$$\begin{aligned} R(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) &= \frac{f_M(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu)}{F_L(\boldsymbol{\Delta\mathcal{I}})} \\ &= \frac{1}{m} \sum_j f_L^{-1}[\Delta\mathcal{I}(j\Delta\nu)] \; f_D(\epsilon_j) \; \Delta\nu \end{aligned} \tag{9}$$

Clearly, the same value of μ maximizes both the marginalized pdf and the likelihood ratio. However, the likelihood ratio is useful not only for parameter estimation, but also for testing the hypothesis that the contour is present.

Taking logarithms, we obtain the log-likelihood ratio:

$$\begin{aligned} h(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) &\stackrel{\text{def}}{=} \log R(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) \\ &= -\log m + \log \left\{ \sum_j f_L^{-1}[\Delta\mathcal{I}(j\Delta\nu)] \; f_D(\epsilon_j) \; \Delta\nu \right\} \end{aligned} \tag{10}$$

which we define as the point-evaluation function. By inserting Eq.1 into Eq.10, and remembering that f_D is Gaussian with variance σ^2, we obtain

$$h(\boldsymbol{\Delta\mathcal{I}} \,|\, \mu) = h_0 + \log \sum_j \exp \sqrt{\frac{|\Delta\mathcal{I}(j\Delta\nu)|}{\lambda}} \; \exp\left(-\frac{\epsilon_j^2}{2\sigma^2} \right) \tag{11}$$

where we define for simplicity:

$$h_0 \stackrel{\text{def}}{=} \log \frac{Z_L}{m} - \log \frac{Z_D}{\Delta\nu} \tag{12}$$

From Eq.11, it can be seen that, for a given observation $\boldsymbol{\Delta\mathcal{I}}$, the point-evaluation becomes larger when the contour is placed at a location μ that maximizes a function of the absolute values $|\Delta\mathcal{I}|$ under a Gaussian window centered on μ. The next section shows that maximization of the point-evaluation can be achieved by iterative minimization (and re-computation) of a squared distance which has greater intuitive appeal than the point-evaluation.

Before deriving the algorithm, however, statistical dependencies between point-evaluations must be considered.

2.4 Correction for Statistical Dependencies

Consider the set of sampled contour locations $\boldsymbol{\mu} \stackrel{\text{def}}{=} \{\mu_k \,|\, 1 \leq k \leq n\}$, where the index k ranges over n observations on all observable line-segments. The number n of observations depends on the number and length of the visible line segments. The sum of all point-evaluations, collected over a finite set of sample points on the projected contours, needs to be corrected for statistical dependencies between different observations, to obtain the log-likelihood of $\boldsymbol{\mu}$.

Statistical parameters of natural images show considerable local variations within a single image (*e.g.* [18]). Therefore, the probability of a large point-evaluation is increased if other large point-evaluations have been observed nearby, independently of whether these large evaluations arise from the contours of interest or are due to the texture of the background.

To a first approximation, dependencies between measurements taken on different line segments can be ignored. Dependencies between measurements taken at different sampling points on the same line segment can be significantly reduced by scaling the corresponding point-evaluation functions by $1/\sqrt{L_k}$, where L_k is the length (in pixels) of the segment including sample point k. This scaling factor has been empirically found to give an approximately constant relationship between log-probabilities and sums of point-evaluations over a single line segment: more details are given in [16]. By this scaling, the objective function to be maximized becomes:

$$H = \sum_{k=1}^{n} \frac{1}{\sqrt{L_k}} \, h(\mu_k \,|\, \boldsymbol{\Delta\mathcal{I}}_k) \tag{13}$$

3 The EM Contour Algorithm

When the log-likelihood is obtained by marginalization, it is often the case that optimization by the EM method leads to more robust convergence, as compared to Newton's method [4,13].

In introducing the algorithm, we begin by considering a single sample point on a contour. Optimization of the object pose under the constraint of rigid object motion on the ground plane is considered in the last subsection.

3.1 Simplified Description

This subsection gives an intuitive motivation for the EM contour algorithm.

Suppose that the true image location ν_b of the boundary between object and background were known. In this case, the image evidence could be ignored, since it would give no additional information: the likelihood of the contour location would be given simply by the likelihood of the deformation:

$$f(\nu_b|\mu) = f_D(\nu_b - \mu) \tag{14}$$

Given that the true location of the boundary is not known, one possible solution is to estimate it from the observation. As an estimator, we choose the *center of mass* of the observation:

$$\widehat{\nu} \stackrel{\text{def}}{=} \sum_j p(j\Delta\nu)\, j\Delta\nu \tag{15}$$

where $p(j\Delta\nu)$ is the probability that the contour is between locations $(j-1/2)\Delta\nu$ and $(j+1/2)\Delta\nu$ on the normal under consideration. This is obtained by normalization to unity of the sum (over all locations on the normal) of the pdf's of the observations:

$$p(j\Delta\nu) = \frac{f(\boldsymbol{\Delta\mathcal{I}}|j\Delta\nu)\ f_D(j\Delta\nu - \mu)}{f(\boldsymbol{\Delta\mathcal{I}}|i\Delta\nu)\ \sum_i f_D(i\Delta\nu - \mu)} \tag{16}$$

The center of mass is the *BLS* (Bayes-least-squares) estimate of the deformation. The use of the *BLS* estimator, rather than *e.g.* the *MAP* estimator, will be justified in the next subsection. For the moment, suffice it to say that it has the advantage of integrating over all the image evidence, rather than using only the image location with the strongest image derivative.

Assuming that the center of mass is a valid estimate, the log-likelihood becomes

$$\log f(\widehat{\nu}|\mu) = \log f_D(\widehat{\nu} - \mu) \tag{17}$$

$$= -\log Z_D - g(\widehat{\nu} - \mu) \tag{18}$$

where we define for convenience

$$g(\widehat{\nu} - \mu) = \frac{(\widehat{\nu} - \mu)^2}{2\sigma^2} \tag{19}$$

The definition of center of mass includes the current estimate of the contour location through Eq. 16. Therefore, an estimate of μ is needed to compute $\widehat{\nu}$, and an estimate of $\widehat{\nu}$ is needed to optimize μ. This fact suggests the following iterative algorithm:

- E step: estimate the deformation $\widehat{\epsilon} = \widehat{\nu} - \mu$ of the contour, using the image evidence $\boldsymbol{\Delta\mathcal{I}}$ and the estimate $\mu^{(t-1)}$ of the contour location (where the superscript $(t-1)$ refers to iteration number);
- M step: re-estimate the contour locations $\mu^{(t)}$ so as to minimize the squared deformation, $g(\widehat{\nu} - \mu)$.

If the minimization is unconstrained, the second step is trivial: just set $\mu^{(t)} = \widehat{\nu}^{(t)}$. Constrained optimization will be considered in subsection 3.3.

3.2 Derivation of the Algorithm

The derivation is analogous to that of the EM clustering algorithm (*e.g.* [13], section 2.7). We begin by writing the expression for the complete-data log-likelihood, *i.e.* the log-pdf of the observation, given the current contour location and the vector of indicator variables $\boldsymbol{\eta}_{\Delta\nu} = \{\eta_{\Delta\nu}(j\Delta\nu)\}$:

$$h_C(\boldsymbol{\Delta\mathcal{I}}|\mu, \boldsymbol{\eta}) = \sum_j \eta_{\Delta\nu}(j\Delta\nu) \log\left[f(\Delta\mathcal{I}|j\Delta\nu)\ f_D(\epsilon_j)\ \Delta\nu\right]$$

$$= h_0 + \sum_j \eta_{\Delta\nu}(j\Delta\nu) \left(\sqrt{\frac{|\Delta\mathcal{I}(j\Delta\nu)|}{\lambda}} - \frac{\epsilon_j^2}{2\sigma^2}\right) \tag{20}$$

Given that $\boldsymbol{\eta}_{\Delta\nu}$ is unknown, we substitute it by its expected value $\widehat{\boldsymbol{\eta}}_{\Delta\nu} = \{p(j\Delta\nu)\}$. By this substitution, we obtain the *expected complete-data log-likelihood*, or *negative free energy*:

$$\widehat{h}_C(\boldsymbol{\Delta\mathcal{I}}|\mu, \widehat{\boldsymbol{\eta}}) = h_0 + \sum_j p(j\Delta\nu) \left(\sqrt{\frac{|\Delta\mathcal{I}(j\Delta\nu)|}{\lambda}} - \frac{\epsilon_j^2}{2\sigma^2}\right) \tag{21}$$

The negative free energy can be re-written to make explicit the dependency on μ:

$$\widehat{h}_C(\boldsymbol{\Delta\mathcal{I}}|\mu,\widehat{\boldsymbol{\eta}}) = h_0 + h_1 - g(\widehat{\nu}-\mu) \tag{22}$$

where $g(\widehat{\nu}-\mu)$ is defined by Eq.19 and

$$h_1 \stackrel{\text{def}}{=} \sum_j p(j\Delta\nu)\left[\sqrt{\frac{|\Delta\mathcal{I}(j\Delta\nu)|}{\lambda}} - \frac{(j\Delta\nu-\widehat{\nu})^2}{2\sigma^2}\right] \tag{23}$$

Once the probabilities $\{p(j\Delta\nu)\}$ are estimated in the E step of the EM algorithm, h_1 is a constant independent of μ.

From Eq.22, it can be seen that the negative free energy $\widehat{h}_C$ is equal to the negative squared distance $-g(\widehat{\nu}-\mu)$ between the contour and the center of mass, plus additive terms which do not depend on μ. We define $g(\widehat{\nu}-\mu)$ as the *differential free energy* that needs to be *minimized* with respect to μ.

A more general definition of the EM contour algorithm, which makes explicit its statistical basis, is as follows: initialize the contour locations by the prediction of a Kalman filter, or whatever initialization method is appropriate; thereafter, iterate to convergence the following steps:

– E step: estimate the posterior pdf over a discrete set of deformations, $p(j\Delta\nu)$, using Eq.16 with the image evidence $\boldsymbol{\mathcal{I}}$ and the current estimate of μ:

$$p^{(t)}(j\Delta\nu) = \frac{f_D(j\Delta\nu-\mu^{(t-1)})\ f(\boldsymbol{\mathcal{I}}|j\Delta\nu)}{\sum_i f_D(i\Delta\nu-\mu^{(t-1)})\ f(\boldsymbol{\mathcal{I}}|i\Delta\nu)} \tag{24}$$

– M step: estimate the contour location

$$\mu^{(t)} = \underset{\mu}{\text{argmax}}\ g^{(t)}(\mu)$$

that maximizes the current estimate of the negative free energy, or equivalently maximizes the current estimate of the free-energy term:

$$\widehat{h}_D^{(t)}(\mu) = \sum_j p^{(t)}(j\Delta\nu)\log f_D(j\Delta\nu-\mu) \tag{26}$$

In this general form, the EM contour algorithm does not involve any explicit estimation of the local shape deformation. Under the assumption of a Gaussian prior pdf for deformations, the EM contour algorithm reduces to the special case outlined in subsection 3.1, in which case the *BLS* estimates of deformations (*i.e.* the centers of mass) can be used for simplicity. In the rest of this paper, we restrict our attention to this special case.

The theory of the EM method ([13], chapter 3) proves that (a) the log-likelihood does not decrease from one iteration to the following and (b) the algorithm converges to an extremum of the log-likelihood.

3.3 Optimization of Pose Parameters

The differential free energy of the entire model can be obtained by substituting f with g in Eq.13:

$$\mathcal{G}(\mathbf{y}) = \sum_{k=1}^{n} \frac{1}{\sqrt{L_k}}\, g\left[\boldsymbol{\Delta\mathcal{I}}_k \,|\, \mu_k(\mathbf{y})\right] \tag{27}$$

where the dependencies on $\mathbf{y}$ have been made explicit for terms on both sides of the equation.

In order to minimize $\mathcal{G}$ in pose-parameter space (*i.e.* under the constraint of rigid translation and rotation of the object on the ground plane), the E and M steps of the EM algorithm must be appropriately modified:

- E step: use full perspective to obtain the current estimates of $\boldsymbol{\mu}$ from the pose parameters $\mathbf{y}^{(t-1)}$, and estimate the posterior pdf's over deformations, $p_k(j\Delta\nu)$, using Eq.16 with the image evidence $\boldsymbol{\Delta\mathcal{I}}_k$ and the current estimate of μ_k;
- M step: estimate the pose parameters $\mathbf{y}^{(t)}$ that minimize the differential free energy (Eq. 27), using the current estimates of the posterior pdf $p_k(j\Delta\nu)$ and linearized inverse perspective.

Since $g(\mu)$ is a quadratic function of μ, it follows that $\mathcal{G}(\mathbf{y})$ is a quadratic function of $\mathbf{y}$ within the range in which the linearized perspective is accurate.

Convergence of the EM contour algorithm depends only on linearized perspective being a valid approximation within the distance between $\mathbf{y}^{(t-1)}$ and $\mathbf{y}^{(t)}$: at the following iteration, full perspective is used to re-estimate the contour locations, the free energy and the linearized perspective.

4 Dynamical Model

In order to track an object over an image sequence, the complete state vector of the vehicle needs to be considered:

$$\mathbf{x} = \{X, Y, \theta_Z, v, \omega_Z, a\} \tag{28}$$

where v is the tangential velocity, ω_Z the angular velocity, and a the tangential acceleration of the object.

The observed pose[2] is equal to the first 3 terms of the state vector, plus an observation noise which includes the combined effect of the shape deformations and image noise:

$$\mathbf{y} = \mathbf{B} \cdot \mathbf{x} + \mathbf{n} \tag{29}$$

where $\mathbf{B}$ is a 3×6 matrix whose elements are all zero, except for $\mathbf{B}_{11} = \mathbf{B}_{22} = \mathbf{B}_{33} = 1$, and $\mathbf{n}$ is the observation noise.

[2] More precisely, the pose estimated by the EM contour algorithm.

In addition to Eq. 29, a dynamical model is needed to predict the state of the object at the next frame. The form of the model is as follows:

$$\mathbf{x}(t+\Delta t) = \mathbf{D}\left[\mathbf{x}(t)\right] + \mathbf{z}(t+\Delta t) \tag{30}$$

where Δt is the time interval between video frames (or, in the case of an iterated Kalman filter, between iteration steps), $\mathbf{D}$ describes the deterministic dynamics of the system and $\mathbf{z}$ is a random vector which includes unknown control inputs from the driver.

Simple physics leads to the following dynamical equations:

– position on the ground plane from orientation and tangential velocity:

$$\begin{aligned} X(t+\Delta t) &= X(t) + v(t)\Delta t \cos\theta_Z(t) \\ Y(t+\Delta t) &= Y(t) + v(t)\Delta t \sin\theta_Z(t) \end{aligned} \tag{31}$$

– orientation and tangential velocity from angular velocity and tangential acceleration:

$$\begin{aligned} \theta_Z(t+\Delta t) &= \theta_Z(t) + \omega_Z(t)\Delta t \\ v(t+\Delta t) &= v(t) + a(t)\Delta t \end{aligned} \tag{32}$$

– angular velocity and tangential acceleration from the driver's input:

$$\begin{aligned} \omega_Z(t+\Delta t) &= \omega_Z(t)[1-\exp(-\Delta t/\tau)] + \mathbf{z}_5(t+\Delta t) \\ a(t+\Delta t) &= a(t)[1-\exp(-\Delta t/\tau)] + \mathbf{z}_6(t+\Delta t) \end{aligned} \tag{33}$$

The simplifying assumption is that changes of pressure on the gas pedal and of steering angle directly translate into tangential acceleration and angular velocity.

Assuming further that these inputs are uncorrelated, the (6×6) covariance matrix $\mathbf{Q}$ of the inputs should have only two non-zero elements: $\mathbf{Q}_{66} = \sigma_a^2$ for the tangential acceleration and $\mathbf{Q}_{55} = \sigma_\omega^2$ for the angular velocity.

In practice, better performance has been obtained by setting

$$\mathbf{Q}_{11} = \sigma_p^2 \sin^2\theta_Z \tag{34}$$

$$\mathbf{Q}_{22} = \sigma_p^2 \cos^2\theta_Z \tag{35}$$

where σ_p^2 is a spurious noise term in the vehicle position, in the direction normal to the vehicle orientation. This term helps to re-align the model with the object after an accidental misalignment.

A convenient measure of the covariance of the observation noise is the inverse of the *empirical observed information matrix* ([13], section 4.3, Eq. 4.41):

$$\boldsymbol{I}_e = \sum_k \frac{1}{\sqrt{L_k}} \nabla h(\boldsymbol{\Delta\mathcal{I}}_k|\mu_k) \cdot \nabla^T h(\boldsymbol{\Delta\mathcal{I}}_k|\mu_k) \tag{36}$$

where the sum is over all observations k and the gradient is in pose-parameter space.

Note that the nonlinearities in the system of Eq. 31 are due to the change from a Cartesian coordinate system (for the vehicle location, X and Y) to a polar coordinate system (for the vehicle orientation and velocity, θ_Z and v). Although the nonlinearities are inconvenient, this change of coordinates is a natural way of enforcing two constraints:

- the orientation of the vehicle θ_Z is directly observable by fitting the vehicle model to the image, as detailed in the previous section, while the vehicle velocity can only be inferred by observing the vehicle pose in two or more frames;
- the vehicle cannot move sideways.

It is also worth pointing out that the nonlinearity of the dynamical model is a lesser problem than the nonlinearity of the observation model.

Having specified the system, it is straightforward to implement an iterated extended Kalman filter [1] for the state variables.

4.1 Parameters of the Method

The observation model has 3 parameters: the scale parameter λ of the pdf of grey level differences; the standard deviation σ of the shape deformations; and the sampling interval $\Delta\nu$ on the normals to the contours.

The parameter λ is estimated from the average square root of gld's $\left\langle \sqrt{\Delta\mathcal{I}} \right\rangle$, measured over the entire image. For $\beta = 0.5$, the maximum-likelihood estimate of λ is given by $\lambda_{ML} = \left\langle \sqrt{\Delta\mathcal{I}} \right\rangle^2 /4$. In the absence of large camera motion or lighting variations, it is only necessary to estimate λ in the first frame of the image sequence.

The scale parameter σ is varied in a range corresponding to a 3-D range of 0.3 m to 0.1 m at the viewing distance of the vehicle being tracked. The EM contour algorithm is applied to convergence at each scale. At each scale σ, the sampling interval is given by $\Delta\nu = \max(1,\ \sigma/4)$ (where both $\Delta\nu$ and σ are in pixels). At each scale, the termination criterion is that the root-mean-square displacement of sample points becomes less than $0.05\,\sigma$.

The parameters of the dynamical model are constrained to a realistic range by the physics of the objects under consideration. The values used in our application are as follows: $\tau = 0.1\ s$; $\sigma_a = 3\ m^2/s$; $\sigma_\omega = 16°/s$; $\sigma_p = 0.5\ m$. The standard deviations have been given relatively high values to compensate for the non-Gaussian nature of the inputs.

5 Initialization

The contour tracker is initialized when a new vehicle is detected and its pose and velocity are estimated. This is accomplished by cluster analysis of image differences. The method is fully described elsewhere [14]. Briefly, a reference image is subtracted from the current image. A new cluster/object is detected when the

image difference has a significant amount of energy at a location distinct from the locations of all currently tracked clusters. The centroid and covariance of clusters are estimated by cluster analysis. When a new cluster is no longer in contact with the image boundaries, its centroid and covariance give initial estimates of the location and size of the corresponding object, under the assumption that the object is on the ground plane. If the estimated size is compatible with a car, then the orientation of the object and its velocity are determined by tracking its position for two frames.

6 Results and Conclusions

The method was tested on the PETS2000 test sequence [5]. The vehicles in the test sequence include a saloon car, a hatchback and a van. The proof of the robustness of the tracker is that the vehicles can be tracked (at 5 frames/second) using a grossly inappropriate geometric model (see Fig. 2), even when they move by almost their full length from one frame to the next. A video showing the tracking results is available at ftp://ftp.diku.dk/pub/diku/image/pece.eccv02.mpg.

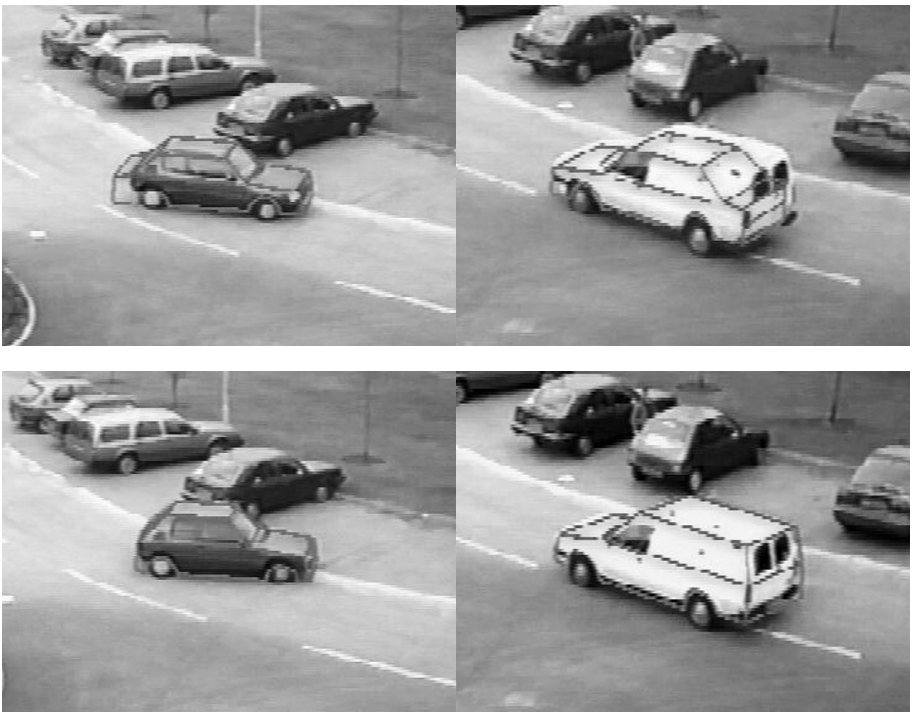

Fig. 2. Tracking of the hatchback and van in the PETS2000 test sequence: the tracker is as effective with a generic vehicle model as with a specific model.

It is interesting that the equations for the log-likelihood are obtained by assuming a uniform pdf of grey-level differences under an object edge and a non-uniform pdf in the absence of an edge. At first sight, this would seem to imply that, paradoxically, the posterior pdf of the data is independent, and the prior pdf of the data dependent, on the state variables. In fact, a careful inspection of Eqs. 4 and 5 shows that this is not the case: the pdf of the image, given the pose variables (Eq.5) does depend on the pose, while the prior pdf of the image (Eq.4) depends on the image, but not on the model, and is not relevant to the optimization task.

One limitation of the method is that, if the object being tracked enters a highly textured region (*e.g.* bushes, cobblestones, other objects in the background), then large gld values will be measured on all normals, independently of the object pose. Whether the object can still be tracked under these conditions depends on how the average grey-level difference between object and background compares to the *local* gld statistics.

By contrast, the method should be relatively robust against occlusion and changes of illumination. In case of partial occlusion, if a few model lines can be reliably located in the image, then these can be sufficient to solve for the object pose, depending on the geometry of these visible lines. In case of changes of illumination, the initialization method can break down (since it is based on *temporal* image differences), but the contour tracker can still work as long as the λ parameter of the pdf of gld's is updated.

The main advantages of the EM contour tracker are its statistical basis in the EM method, its operation in 3-D and the two simple concepts on which the tracker is based: there are no correlations between image pixels on opposite sides of object edges, and it is necessary to marginalize over shape variations.

The fact that the generative model leads to a simple marginalization technique is an attractive feature. Methods that involve feature detection at any stage should marginalize 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 should marginalize 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. In practice, such marginalization is often difficult: avoiding feature detection makes marginalization easier to implement.

References

1. Y. Bar-Shalom, T. E. Fortmann, *Tracking And Data Association.* Academic Press, 1988.
2. A. Baumberg, D.C. Hogg, Learning Flexible Models from Image Sequences, Proc. of ECCV'94, Lecture Notes in Computer Science, vol. 800, *pp.*299-308, 1994.
3. A Blake, M Isard, *Active Contours*, Springer-Verlag, Berlin 1998.
4. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion), *J. of the Royal Statistical Soc.* B 39: 1-38, 1977.

5. JM Ferryman, AD Worrall, Proc. of the 1st IEEE International Workshop on Performance Evaluation in Tracking and Surveillance: PETS 2000.
6. D Geman, B Jedynak, An active testing model for tracking roads in satellite images. IEEE Trans. PAMI 18(1): 1-14, 1996.
7. C Harris, Tracking with rigid models. In *Active Vision* (A Blake, A Yuille, eds) *pp.*59-73. MIT Press, 1992.
8. J. Huang, D. Mumford, Statistics of natural images and models. In *Proc. CVPR'99*, vol.1, *pp.*541-547, 1999.
9. A. Lee, D. Mumford, An Occlusion Model generating Scale-Invariant Images. In *Proc. Int. Workshop on Statistical and Computational Theories of Vision.* http://www.cis.ohio-state.edu/ szhu/SCTV2001.html, 1999.
10. M Kass, A Witkin, D Terzopoulos, Snakes: active contour models. In *Proc. ICCV'97*, *pp.*259-268, 1987.
11. 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.
12. H Kollnig, H-H Nagel, 3D pose estimation by fitting image gradients directly to polyhedral models. In *Proc. ICCV'95*, *pp.*569-574, 1995.
13. G.J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.
14. A.E.C. Pece, Generative-model-based tracking by cluster analysis of image differences. Accepted for publication in *Robotics and Autonomous Systems.*
15. AEC Pece, AD Worrall, A statistically-based Newton method for pose refinement. *Image and Vision Computing* 16: 541-544, 1998.
16. AEC Pece, AD Worrall, A Newton method for pose refinement of 3D models. Proc 6th International Symposium on Intelligent Robotic Systems: SIRS'98, The University of Edinburgh.
17. A.E.C. Pece, A.D. Worrall, Tracking without feature detection. In [5], *pp.*29-37, 2000.
18. DL Ruderman, The statistics of natural images. *Network* 5: 517-548, 1994.
19. H. Sidenbladh, M.J. Black, Learning image statistics for Bayesian tracking, In *Proc. ICCV'01*, Vol. 2, *pp.*709-716, 2001.
20. P. Tissainayagam, D. Suter, Tracking multiple object contour with automatic motion model switching, In *Proc. ICPR'00*, *pp.*1146-1149, 2000.
21. AD Worrall, JM Ferryman, GD Sullivan, KD Baker, Pose and structure recovery using active models. Proc. BMVC'95, *pp.*137-146, 1995.
22. AL Yuille, JM Coughlan, Fundamental limits of Bayesian inference: order parameters and phase transitions for road tracking. *IEEE Trans. PAMI* 22(2): 160-173, 2000.