Robust Real-Time Visual Tracking Using Pixel-Wise Posteriors*

Charles Bibby and Ian Reid

Active Vision Lab
Department of Engineering Science
University of Oxford
cbibby@robots.ox.ac.uk,ian@robots.ox.ac.uk

Abstract. We derive a probabilistic framework for robust, real-time, visual tracking of previously unseen objects from a moving camera. The tracking problem is handled using a bag-of-pixels representation and comprises a rigid registration between frames, a segmentation and online appearance learning. The registration compensates for rigid motion, segmentation models any residual shape deformation and the online appearance learning provides continual refinement of both the object and background appearance models. The key to the success of our method is the use of pixel-wise posteriors, as opposed to likelihoods. We demonstrate the superior performance of our tracker by comparing cost function statistics against those commonly used in the visual tracking literature. Our comparison method provides a way of summarising tracking performance using lots of data from a variety of different sequences.

1 Introduction

Fast and reliable visual tracking is a prerequisite for a vast number of applications in computer vision. Though it has been the subject of intense effort over the last two decades, it remains a difficult problem for a number of reasons. In particular, when tracking previously unseen objects, many of the constraints that give reliability to other tracking systems – such as strong prior information about shape, appearance or motion – are unavailable.

One technique that has shown considerable promise for its ability to perform tracking and segmentation within a unified framework is the use of an implicit contour, or level-set to represent the boundary of the target [1,2,3]. As well as handling topological changes seamlessly, tracking using level-sets can be couched in a fairly standard probabilistic formulation [4,5], and hence can leverage the power of Bayesian methods.

In this paper, we present a novel probabilistic framework for combined tracking and segmentation, which, as well as capturing all of the desirable properties of level-set based tracking, is very robust and runs in a few milliseconds on standard hardware. We base our framework on a generative model of image formation that represents the image as a bag-of-pixels [6]. The advantage of such

* We gratefully acknowledge the sponsorship of Servowatch Systems Ltd.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part II, LNCS 5303, pp. 831–844, 2008.

a model – in common with other simpler density-based representations such as colour-histograms – is the degree of invariance to viewpoint this confers.

Like [4], we derive a probabilistic, region based, level-set framework, which comprises an optimal rigid registration, followed by a segmentation to re-segment the object and account for non-rigid deformations. Aside from issues of speed (which are not addressed in [4]) there are a number of key differences between [4] and our work, some of which stem from the generative model we use for image data (see Sect. 2). First, our derivation gives a probabilistic interpretation to the Heaviside step function used in most region based level-set methods [7,4]. Second, given this interpretation we propose a pixel-wise posterior term, as opposed to a likelihood, which allows us to marginalise out model parameters at a pixel level. As we show in Sect. 2, this derives naturally from our generative model, and is a subtle but absolutely crucial difference between our method and others e.g. [4,2,3], as our results show in Sect. 7. Third, in contrast to [7,4] and similar to [8,9] we assume a non-parametric distribution for image values as opposed to a single Gaussian (for an entire region). Finally, we introduce a prior on the embedding function which constrains it to be an approximate signed distance function. We show that this gives a clean probabilistic interpretation to the idea proposed by [10] and avoids the need for reinitialisation of the embedding function that is necessary in the majority of level-set based approaches.

Our work also bears some similarity to [11] who sought the rigid transformation that best aligns a fixed shape-kernel with image data using the Bhattacharyya coefficient. This work extended the pioneering work of this type [12,13] to handle translation+scale+rotation as opposed to translation only or translation+scale. In contrast to [11], however, we allow the shape to change online and propose a novel framework using pixel-wise posteriors, which removes the cost of building an empirical distribution and testing it with the Bhattacharyya coefficient. This has a second hidden benefit as it avoids the need to build a 'good' empirical distribution given limited data; we find in practice this gives a significant improvement over [12,13,11].

Unlike [4], [8,9] use a non-parametric distribution for image data. They derive contour flows based on both KL-divergence and the Bhattacharyya coefficient. Though they demonstrate that both are effective for tracking, they do not model rigid transformation parameters explicitly, they must recompute their non-parametric distributions at every iteration, and – as we show in Sect. 7 – objectives based on the Bhattacharyya coefficient are inferior to the one we propose.

Finally, it is worth mentioning template based tracking methods (see [14] for an excellent summary of past work). We include an ideal SSD cost in our comparative results (Sect. 7), which uses the correct template at each frame. Though this unfairly advantages the SSD method – since in reality the exact template is *never* available – it does suggest that in future there would be benefit in considering how spatial information can be incorporated. Thus although our method is currently based solely on non-parametric distributions we are currently investigating ways to augment it with spatial information.

The key problem with template tracking is how to adapt the template over time. Within our framework (and other similar work), because the segmentation is performed rapidly and reliably online, the appearance and shape models of the object can be updated over time without suffering from the significant problems of drift that plague other algorithms.

Our framework is general enough to be extended to various types of prior information and various imaging modalities, but in this paper we restrict ourselves to the problem of tracking the 2D projections of either 2D or 3D objects in ordinary colour video. In summary, the key benefits of our method are: (i) an extensible probabilistic framework; (ii) robustness - given by pixel-wise posteriors and marginalisation; (iii) real-time performance; (iv) excellent cost function characteristics; (v) no need to compute empirical distributions at every frame; (vi) online learning (i.e. adaption of appearance and shape characteristics); (vii) flexibility to track many different types of object and (viii) high invariance to view and appearance changes.

The remainder of this paper is organised as follows: Sect. 2 describes the representation of the object being tracked and derives a probabilistic framework from a simple generative model; Sect. 3 outlines the level-set segmentation; Sect. 4 shows the registration process; Sect. 5 describes our method for dealing with drift; Sect. 6 outlines the online learning process; Sect. 7 shows our results and Sect. 8 concludes with a summary and discussion.

2 The Generative Model

We represent the object being tracked by its shape $\mathbf{C}$, its location in the image $\mathbf{W}(\mathbf{x}, \mathbf{p})$ and two underlying appearance models: one for the foreground $P(\mathbf{y}|M_f)$ and one for the background $P(\mathbf{y}|M_b)$. Figure 1 illustrates this with a simple example.

Shape. is represented by the zero level-set $\mathbf{C} = \{\mathbf{x}|\mathbf{\Phi}(\mathbf{x}) = 0\}$ of an embedding function $\mathbf{\Phi}(\mathbf{x})$ [1,5]. The pixels $\mathbf{\Omega}$ in the object frame are segmented into two regions: one for the foreground $\mathbf{\Omega}_f$ and one for the background $\mathbf{\Omega}_b$.

Location. is described by a warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ that takes a pixel location $\mathbf{x}$ in the object frame and warps it into the image frame according to parameters $\mathbf{p}$. This warp must form a group [14]; however, this is acceptable as many common useful transformations in computer vision do form groups, for instance: translation, translation+scale, similarity transforms, affine transforms and homographies.

Appearance models. $P(\mathbf{y}|M_f)$ and $P(\mathbf{y}|M_b)$ are represented with YUV histograms using 32 bins per channel. The histograms are initialised either from a detection module or a user inputted initial bounding box. The pixels inside the bounding box are used to build the foreground model and the pixels from an inflated bounding box are used to build the background model. The two initial distributions are then used to produce a tentative segmentation, which is in turn used to rebuild the model. This procedure is iterated until the shape converges

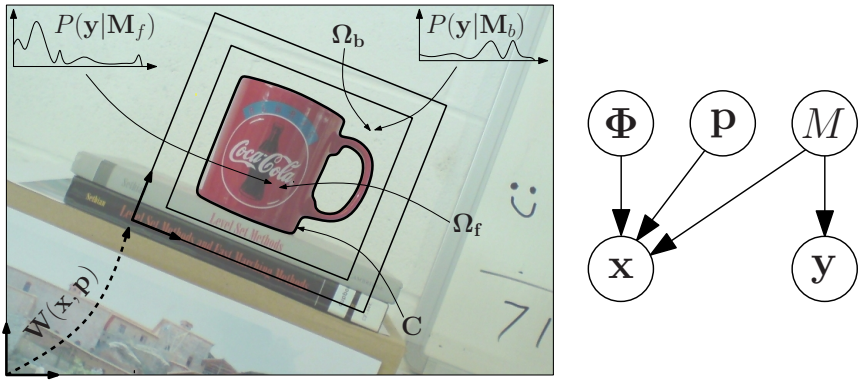

Fig. 1. (Left): Representation of object showing: the contour $\mathbf{C}$, the set of foreground pixels $\mathbf{\Omega}_f$, the set of background pixels $\mathbf{\Omega}_b$, the foreground model $P(\mathbf{y}|M_f)$, the background model $P(\mathbf{y}|M_b)$ and the warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$; (Right): Graphical representation of our generative model representing the image as a bag-of-pixels

(similar to [15]). Once tracking commences the appearance models and shape $\mathbf{C}$ are estimated (adapted) online, as described in Sect. 6. In summary, we use the following notation:

- $\mathbf{x}$: A pixel location in the object coordinate frame.
- $\mathbf{y}$: A pixel value (in our experiments this is a YUV value).
- $\mathbf{I}$: Image.
- $\mathbf{W}(\mathbf{x}, \mathbf{p})$: Warp with parameters $\mathbf{p}$.
- $M = \{M_f, M_b\}$: Model parameter either foreground or background.
- $P(\mathbf{y}|M_f)$: Foreground model over pixel values $\mathbf{y}$.
- $P(\mathbf{y}|M_b)$: Background model over pixel values $\mathbf{y}$.
- $\mathbf{C}$: The contour that segments the foreground from background.
- $\mathbf{\Phi}(\mathbf{x})$: Shape kernel (in our case the level-set embedding function).
- $\mathbf{\Omega} = \{\mathbf{\Omega}_f, \mathbf{\Omega}_b\}$: Pixels in the object frame $[\{\mathbf{x}_0, \mathbf{y}_0\}, \ldots, \{\mathbf{x}_N, \mathbf{y}_N\}]$, which is partitioned into foreground pixels $\mathbf{\Omega}_f$ and background pixels $\mathbf{\Omega}_b$.
- $H_\epsilon(z)$: Smoothed Heaviside step function.
- $\delta_\epsilon(z)$: Smoothed Dirac delta function.

Figure 1 illustrates the simple generative model we use to represent the image formation process. This model treats the image as a bag-of-pixels [6] and can, given the model M, the shape $\mathbf{\Phi}$ and the location $\mathbf{p}$, be used to sample pixels $\{\mathbf{x}, \mathbf{y}\}$. Although the resultant image would not look like the true foreground/background image to a human (the pixels would be jumbled up), the colour distributions corresponding to the foreground/background regions $\mathbf{\Omega}_f/\mathbf{\Omega}_b$ would match the models $P(\mathbf{y}|M_f)$ and $P(\mathbf{y}|M_b)$. It is this simplicity that gives more invariance to viewpoint and allows 3D objects to be tracked robustly without having to model their specific 3D structure. The joint distribution for a single pixel given by the model in Fig. 1 is:

$$P(\mathbf{x}, \mathbf{y}, \mathbf{\Phi}, \mathbf{p}, M) = P(\mathbf{x}|\mathbf{\Phi}, \mathbf{p}, M)P(\mathbf{y}|M)P(M)P(\mathbf{\Phi})P(\mathbf{p}). \quad (1)$$

We now divide (1) by $P(\mathbf{y}) = \sum_M P(\mathbf{y}|M)P(M)$ to give:

$$P(\mathbf{x}, \mathbf{\Phi}, \mathbf{p}, M|\mathbf{y}) = P(\mathbf{x}|\mathbf{\Phi}, \mathbf{p}, M)P(M|\mathbf{y})P(\mathbf{\Phi})P(\mathbf{p}), \tag{2}$$

where the term $P(M|\mathbf{y})$ is the pixel-wise posterior, of the models M, given a pixel value $\mathbf{y}$:

$$P(M_j|\mathbf{y}) = \frac{P(\mathbf{y}|M_j)P(M_j)}{\sum_{\{i=f,b\}} P(\mathbf{y}|M_i)P(M_i)} \qquad j = \{f, b\}. \tag{3}$$

Using this posterior is equivalent to applying Bayesian model selection to each individual pixel.[1] We now marginalise over the models M, yielding the pixel-wise posterior probability of the shape $\mathbf{\Phi}$ and the location $\mathbf{p}$ given a pixel $\{\mathbf{x}, \mathbf{y}\}$:

$$P(\mathbf{\Phi}, \mathbf{p}|\mathbf{x}, \mathbf{y}) = \frac{1}{P(\mathbf{x})} \sum_{\{i=f,b\}} \{P(\mathbf{x}|\mathbf{\Phi}, \mathbf{p}, M_i)P(M_i|\mathbf{y})\} P(\mathbf{\Phi})P(\mathbf{p}). \tag{4}$$

Note that the pixel-wise posterior and marginalisation are the subtle but crucial differences to the work in [4], which lacks the marginalisation step and uses a pixel-wise likelihood $P(\mathbf{y}|M)$. We show in Sect. 7 that our formulation yields a much better behaved objective. We consider two possible methods for fusing the pixel-wise posteriors: (i) a logarithmic opinion pool (LogOP):

$$P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega}) = \prod_{i=1}^{N} \left\{ \sum_M \{P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, M)P(M|\mathbf{y}_i)\} \right\} P(\mathbf{\Phi})P(\mathbf{p}) \tag{5}$$

and (ii) a linear opinion pool (LinOP):

$$P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega}) = \sum_{i=1}^{N} \left\{ \sum_M \{P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, M)P(M|\mathbf{y}_i)\} \right\} P(\mathbf{\Phi})P(\mathbf{p}). \tag{6}$$

The logarithmic opinion pool is normally the preferred choice and is most similar to previous work [4,5]; whereas the linear opinion pool is equivalent to marginalising over the pixel locations – this is allowed as our bag-of-pixels generative model treats pixel locations as a random variable. We continue our derivation assuming a logarithmic opinion pool for clarity, but also include results using a linear opinion pool for completeness. Note the term $\frac{1}{P(\mathbf{x})}$ has been dropped as it is constant for all pixel locations and we only seek to maximise $P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})$.

3 Segmentation

The typical approach to region based segmentation methods is to take a product of the pixel-wise likelihood functions $\prod_{i=1}^{N} P(\mathbf{I}(x_i)|M_i)$, over pixel locations $\mathbf{x}_i$,

[1] It is also the distribution that would be computed in the E-step of an EM solution to the problem.

to get the overall likelihood $P(\mathbf{I}|M)$. This can then be expressed as a summation by taking logs and optimised using variational level-sets [1,5]. In contrast to these methods, our derivation leads to pixel-wise posteriors and marginalisation (5), a subtle but important difference.

For the remainder of this section, in order to simplify our expressions (and without loss of generality), we assume that the registration is correct and therefore $\mathbf{x}_i = \mathbf{W}(\mathbf{x}_i, \mathbf{p})$. We now specify the term $P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, M)$ in (5) and the term $P(M)$ in (3) :

$$P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, M_f) = \frac{H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))}{\eta_f} \qquad P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, M_b) = \frac{1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))}{\eta_b} \tag{7}$$

$$P(M_f) = \frac{\eta_f}{\eta} \qquad P(M_b) = \frac{\eta_b}{\eta}, \tag{8}$$

where

$$\eta = \eta_f + \eta_b, \qquad \eta_f = \sum_{i=1}^{N} H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)), \qquad \eta_b = \sum_{i=1}^{N} 1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)). \tag{9}$$

Equation (7) represents normalised versions of the blurred Heaviside step functions used in typical region based level-set methods and can now be interpreted probabilistically as model specific spatial priors for a pixel location $\mathbf{x}$. Equation (8) represents the model priors, which are given by the ratio of the area of the model specific region to the total area of both models. Equation (9) contains the normalisation constants (note that $\eta = N$).

We now specify a geometric prior on $\mathbf{\Phi}$ that rewards a signed distance function:

$$P(\mathbf{\Phi}) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(|\nabla \mathbf{\Phi}(\mathbf{x}_i)| - 1)^2}{2\sigma^2}, \tag{10}$$

where σ specifies the relative weight of the prior. This gives a probabilistic interpretation to the work in [10]. Substituting (7), (8), (9) and (10) into (5) and taking logs, gives the following expression for the log posterior:

$$\log(P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})) \propto \sum_{i=1}^{N} \left\{ \log\left(P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i)\right) - \frac{(|\nabla \mathbf{\Phi}(\mathbf{x}_i)| - 1)^2}{2\sigma^2} \right\} + N \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) + \log(P(\mathbf{p})), \tag{11}$$

where

$$P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i) = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))P_f + (1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)))P_b$$

and

$$P_f = \frac{P(\mathbf{y}_i|M_f)}{\eta_f P(\mathbf{y}_i|M_f) + \eta_b P(\mathbf{y}_i|M_b)} \qquad P_b = \frac{P(\mathbf{y}_i|M_b)}{\eta_f P(\mathbf{y}_i|M_f) + \eta_b P(\mathbf{y}_i|M_b)}.$$

Given that we are about to optimise w.r.t to $\mathbf{\Phi}$ we can drop the last two terms in (11) and by calculus of variations [16] express the first variation (Gateaux derivative) of the functional as:

$$\frac{\partial P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})}{\partial \mathbf{\Phi}} = \frac{\delta_\epsilon(\mathbf{\Phi})(P_f - P_b)}{P(\mathbf{x}|\mathbf{\Phi}, \mathbf{p}, \mathbf{y})} - \frac{1}{\sigma^2}\left[\nabla^2\mathbf{\Phi} - \text{div}\left(\frac{\nabla\mathbf{\Phi}}{|\nabla\mathbf{\Phi}|}\right)\right], \tag{12}$$

where ∇^2 is the Laplacian operator and $\delta_\epsilon(\mathbf{\Phi})$ is the derivative of the blurred Heaviside step function, i.e. a blurred Dirac delta function. Interestingly, $\delta_\epsilon(\mathbf{\Phi})$ can now be interpreted as a way of expressing uncertainty on the contour $\mathbf{C}$. If we were to use Gaussian uncertainty for the contour then the region based uncertainty would be expressed in terms of $\text{erf}(\mathbf{\Phi})$ instead of $H_\epsilon(\mathbf{\Phi})$ (we do not explore this any further in this paper). We seek $\frac{\partial P(\mathbf{\Phi}, \mathbf{p}||\mathbf{\Omega})}{\partial \mathbf{\Phi}} = 0$ by carrying out steepest-ascent using the following gradient flow:

$$\frac{\partial P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})}{\partial t} = \frac{\partial P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})}{\partial \mathbf{\Phi}}. \tag{13}$$

In practice this is implemented using a simple numerical scheme on a discrete grid. All spatial derivatives are computed using central differences and the Laplacian uses a 3×3 spatial kernel. We use $\sigma = \sqrt{50}$ and a timestep $\tau = 1$ for all experiments. For stability $\frac{\tau}{\sigma^2} < 0.25$ must be satisfied (see [10] for details).

4 Tracking

It is possible to pose the tracking problem directly in a segmentation framework [8]. Instead, like [4] we model the frame-to-frame registration explicitly, by having the level-set in the object frame and introducing a warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ into (11). The main benefits of this approach are: (i) control over the interaction between registration (tracking) and segmentation (local shape deformation); (ii) by registering the embedding function first, fewer iterations are required to take account of shape changes (in fact we find one per frame is adequate for our sequences). We now drop any terms in (11) that are not a function of $\mathbf{p}$ in preparation for differentiation:

$$\log(P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})) \propto \sum_{i=1}^{N}\Big\{\log\left(P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i)\right)\Big\} + \log(P(\mathbf{p})) + const. \tag{14}$$

Introducing a warp $\mathbf{W}(\mathbf{x}_i, \Delta\mathbf{p})$ into (14) and dropping the prior term for brevity (we revisit this term in Sect. 5):

$$\log(P(\mathbf{\Phi}, \mathbf{p}|\mathbf{\Omega})) \propto \sum_{i=1}^{N}\Big\{\log\left(P(\mathbf{W}(\mathbf{x}_i, \Delta\mathbf{p})|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i)\right)\Big\}, \tag{15}$$

where $\Delta\mathbf{p}$ represents an incremental warp of the shape kernel. There are many ways this expression could be optimised, the most similar work uses simple

gradient ascent [4]. In contrast, we take advantage of the fact that all of the individual terms are probabilities, and therefore strictly positive. This allows us to write certain terms as squared square-roots and substitute in a first-order Taylor series approximation for each square-root, for example:

$$\left[\sqrt{H_\epsilon(\mathbf{\Phi}(\mathbf{W}(\mathbf{x}_i, \Delta\mathbf{p}))}\right]^2 \approx \left[\sqrt{H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))} + \frac{1}{2\sqrt{H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))}}\mathbf{J}\Delta\mathbf{p}\right]^2, \tag{16}$$

where:

$$\mathbf{J} = \frac{\partial H_\epsilon}{\partial \mathbf{\Phi}}\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}}\frac{\partial \mathbf{W}}{\partial \Delta\mathbf{p}} = \delta_\epsilon(\mathbf{\Phi}(\mathbf{x}_i)) \bigtriangledown\mathbf{\Phi}(\mathbf{x}_i) \frac{\partial \mathbf{W}}{\partial \Delta\mathbf{p}}.$$

Likewise we apply a similar expansion to $(1 - H_\epsilon(\mathbf{\Phi}(\mathbf{W}(\mathbf{x}_i, \Delta\mathbf{p})))$, allowing us then to optimise using Gauss-Newton[2]. This has the advantage that the Hessian itself is not required, rather, a first-order approximation of the Hessian is used. In consequence it is fast, and in our experience exhibits rapid and reliable convergence in our problem domain. It also avoids the issues highlighted in [17] of choosing the appropriate step size for gradient ascent. Excluding the full details for brevity we arrive at an expression for $\Delta\mathbf{p}$:

$$\Delta\mathbf{p} = \left[\sum_{i=1}^{N} \frac{1}{2P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i)} \left(\frac{P_f}{H(\mathbf{\Phi}(\mathbf{x}_i))} + \frac{P_b}{(1 - H(\mathbf{\Phi}(\mathbf{x}_i)))}\right) \mathbf{J}^T\mathbf{J}\right]^{-1} \times \sum_{i=1}^{N} \frac{(P_f - P_b)\mathbf{J}^T}{P(\mathbf{x}_i|\mathbf{\Phi}, \mathbf{p}, \mathbf{y}_i)}. \tag{17}$$

Equation (17) is then used to update the parameters $\mathbf{p}$ by composing $\mathbf{W}(\mathbf{x}_i, \mathbf{p})$ with $\mathbf{W}(\mathbf{x}_i, \Delta\mathbf{p})^{-1}$, analogous to inverse compositional tracking [14].

5 Drift Correction

Having the object represented by its location $\mathbf{p}$ and shape $\mathbf{\Phi}$ leaves an ambiguity where it is possible to explain rigid transformations of the shape either with $\mathbf{p}$ or $\mathbf{\Phi}$. Ideally, any rigid motion would be explained solely by $\mathbf{p}$; however, over time the shape $\mathbf{\Phi}$ slowly incorporates a rigid transformation. We define a prior on the location $P(\mathbf{p})$ which makes small corrections to keep the left/right and top/bottom borders[3] balanced and the minimum border distance equal to four pixels. This is implemented using a proportional controller that takes the four border distances as its input and outputs the prior $P(\mathbf{p})$.

[2] The Taylor expansion is poorly conditioned if $H_\epsilon(\mathbf{\Phi}(\mathbf{x}_i) = 0$; in practice this does not happen as the terms are never equal to zero.

[3] Smallest distances between the contour and the corresponding side of the foreground box.

6 Online Learning

Once registration and segmentation are completed both the foreground and background models are adapted online. This is achieved using linear opinion pools with variable learning rates α_i, $i = \{f, b\}$:

$$P_t(\mathbf{y}|M_i) = (1 - \alpha_i)P_{t-1}(\mathbf{y}|M_i) + \alpha_i P_t(\mathbf{y}|M_i), \qquad i = \{f, b\}. \tag{18}$$

In all experiments $\alpha_f = 0.02$ and $\alpha_b = 0.025$. For shape adaptation we control the evolution rate of the level-set using the timestep τ. Ideally these three parameters would change dynamically throughout the sequence to prevent learning occurring during times of confusion or if the object is lost; we intend to do this in future work.

7 Results

We have tested our system extensively on live video, as well as on a variety of recorded sequences which include objects that exhibit rapid and agile motion with significant motion blur, varying lighting, moving cameras, and cluttered and changing backgrounds. Figure 2 shows a qualitative evaluation of our method on three sequences. The first is a speedboat undergoing a 180° out-of-plane rotation – note how the shape is adapted online. The second is a person jumping around – note the motion blur and shape adaptation. Finally, the third is a hand being tracked from a head mounted camera past a challenging background which has a similar appearance to the object.

To perform a quantitative evaluation we have analysed the characteristics of the underlying cost function of our technique and compared this with competing alternatives, on a set of pre-recorded video sequences. Figure 3 shows still images taken mid-sequence from a subset of these sequences; the minimum length is 400 frames and the total number of frames is over 20,000. To facilitate visualisation of the results we use a 2D rigid transformation + scale, considering each of the four dimensions separately. The competing cost functions considered correspond to the following alternative methods of tracking: level-set methods based on likelihoods [4,2], mean-shift [12,13,11], inverse compositional [14] and distribution based tracking [8,9].

A good cost function has a single extremum at the true location. A poor one has multiple extrema and any local optimisation technique is liable to fall into one of these, which in practice is often the start of tracking failure. For each video frame and each dimension (translation in x and y, rotation and scale) we compute the objectives for the competing cost functions at 40 evenly spaced points over an interval centred at the true state. We then extract all local extrema from these objectives and examine how they are distributed across the interval. To summarise this information we compute a distribution for each dimension and each cost function, using our collection of over 20,000 frames. The ideal distribution would be a delta function centred on the true state; whereas a good distribution would be peaky around the true state and have low probability of local extrema within the region it will be required to converge from. A bad

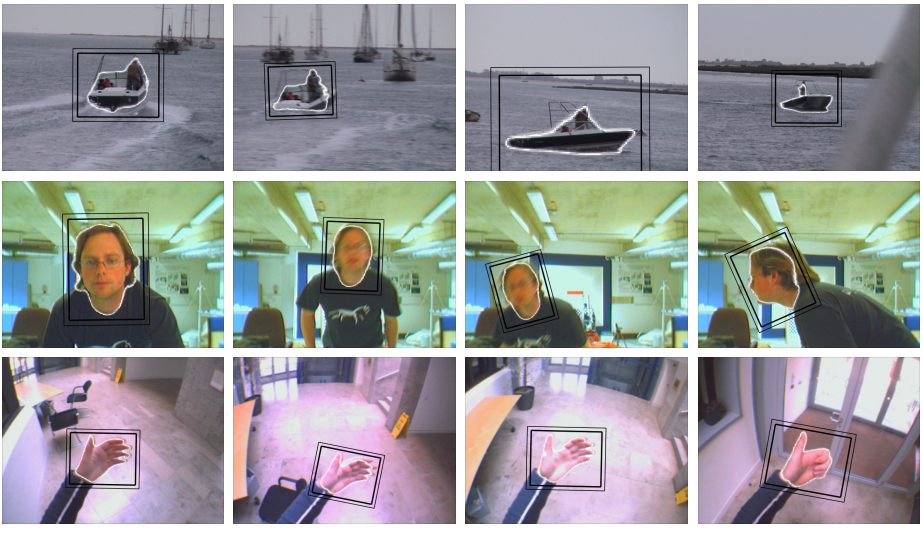

Fig. 2. Qualitative evaluation: (top) a speedboat undergoing a 180° out-of-plane rotation illustrating shape adaptation; (middle) a person jumping around with significant motion blur; (bottom) a hand being tracked in front of a challenging background

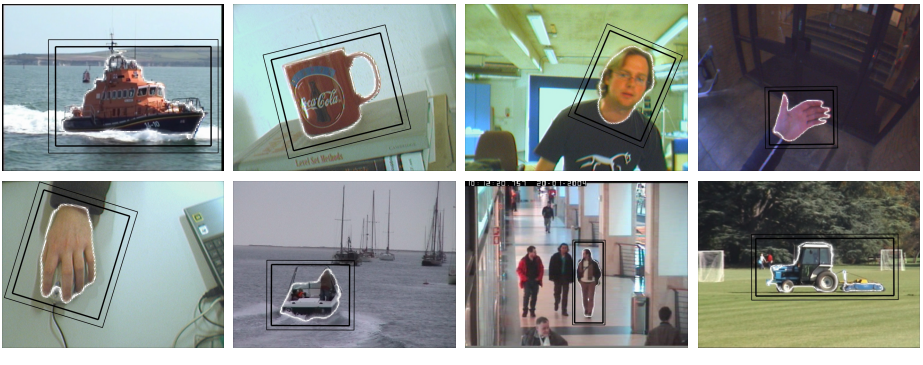

Fig. 3. A selection of video frames from the data sets: (1st row) lifeboat, Coca-Cola mug, a face and a hand filmed from a head mounted camera; (2nd row) a hand using a mouse, a speedboat, a person from the caviar data set [18] and a tractor. The white indicates the current segmentation and the two black boxes indicate the object coordinate frame.

distribution would be relatively flat with high probability of local extrema over the entire space. The particular cost functions we consider are:

- **LogPWP:** Pixel-wise posteriors fused using a logarithmic opinion pool.
- **LinPWP:** Pixel-wise posteriors fused using a linear opinion pool.
- **LogLike:** Log likelihood, used in most level-set work [5,4,2].

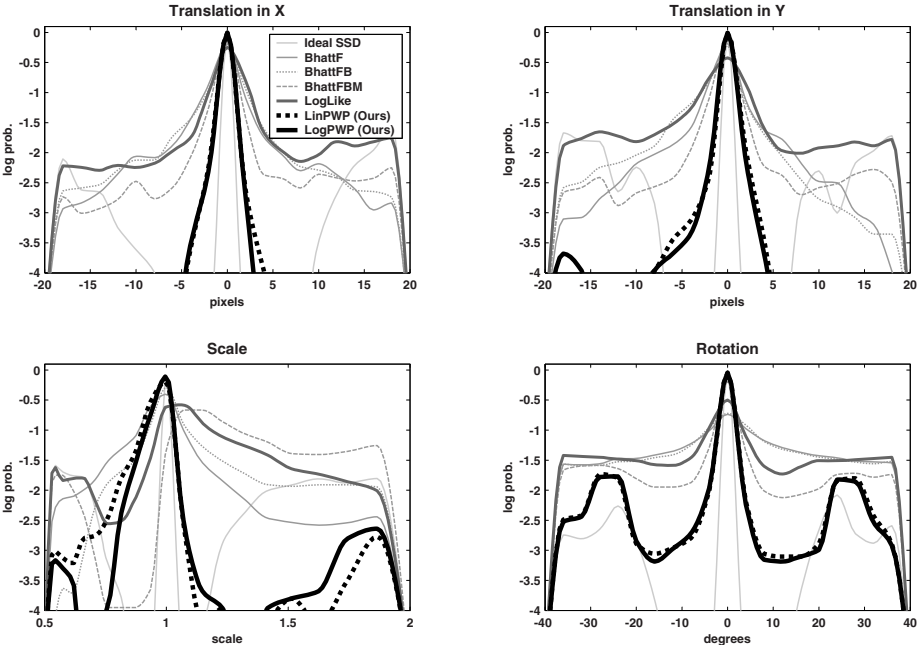

Fig. 4. Quantitative Analysis: Log probability distribution of extrema in the cost functions generated from 20,000 frames of real video data

- **BhattF:** Bhattacharyya coefficient: $B(\mathbf{\Omega}_f) = \sum_{j=1}^{V} \sqrt{P(\mathbf{y}_j|M_f)P(\mathbf{y}_j|\mathbf{\Omega}_f)}$, used by [12,13,11].
- **BhattFB:** Bhattacharyya coefficient with a background model: $B(\mathbf{\Omega}_f, \mathbf{\Omega}_b) = \sum_{j=1}^{V} \sqrt{P(\mathbf{y}_j|M_f)p(\mathbf{y}_j|\mathbf{\Omega}_f)} + \sum_{j=1}^{V} \sqrt{P(\mathbf{y}_j|M_b)P(\mathbf{y}_j|\mathbf{\Omega}_b)}$.
- **BhattFBM:** Bhattacharyya coefficient with a background mismatch: $B(\mathbf{\Omega}_f, \mathbf{\Omega}_b) = \sum_{j=1}^{V} \sqrt{P(\mathbf{y}|M_f)p(\mathbf{y};\mathbf{\Omega}_f)} - \sum_{j=1}^{V} \sqrt{P(\mathbf{y}_j|M_f)P(\mathbf{y}_j|\mathbf{\Omega}_b)}$, suggested by [9].
- **Ideal SSD:** Sum of squared pixel differences using the ideal template i.e. the template extracted at the current location $\mathbf{p}$. This is essentially what you would get if you had the perfect generative model giving the true pixel value at each pixel location including the noise. This of course is *never* going to be achievable but has been included as a useful benchmark and gives an indication of what effect incorporating texture may have.

Note: V is the number of pixel values i.e. $32 \times 32 \times 32$; $P(\mathbf{y}|\mathbf{\Omega}_i)$ $i = \{f, b\}$ is the empirical density built from the pixels $\mathbf{\Omega}_i$ and when computing Bhattacharyya coefficients we weight the contribution of each pixel according to our shape kernel, which is identical to Yilmaz's work [11].

Figure 4 shows distributions generated from over 20,000 real video frames for: translation in x, translation in y, scale and rotation.

- **Translation in x and y:** Our method has narrower distributions near the true state than all methods apart from ideal SSD and is significantly better than the log likelihood used by [4]. Unlike the other methods it also exhibits virtually no extrema outside a ± 5 pixel region – this means that our methods will converge to within ± 5 pixels of the true state from anywhere within the ± 20 pixel space we have evaluated.
- **Scale:** The Bhattacharyya method and Bhattacharyya with background mismatch both have poor localisation in scale, which is in agreement with the findings of many authors. The log likelihood also poorly localises scale compared with our posterior based methods.
- **Rotation:** All Bhattacharyya methods and the log likelihood are poor at correctly localising the rotation. The straight Bhattacharyya coefficient for example has more than a 1% chance of exhibiting extrema anywhere in the rotation space, at a 30Hz frame rate this corresponds to approximately 1 frame in every 3 seconds of video. It is worth noting that the side lobes (at approximately 25°) exhibited by our methods and ideal SSD are due to the self similarity corresponding to fingers in the hand sequences.

Experimentally we were unable to make the log likelihood successfully track several of our sequences, which is confirmed by its poor performance in Fig. 4. One possible explanation is that in other work [4,5,2], a single Gaussian parametric model is used. This implicitly enforces a smooth, unimodal distribution for the joint likelihood. Non-parametric representations do not exhibit these properties; however, they are better at describing complicated distributions and therefore desirable. The reason that our method can deal with these distributions is because of the normalising denominator in (3) and the marginalisation step in (4). These two steps prevent individual pixels from dominating the cost function hence making it smoother and well-behaved.

The work of [8] and its subsequent improvement [9] use distribution matching techniques to incorporate non-parametric distributions into a level-set framework. These methods, similar to the Bhattacharyya based methods, involve computing the empirical densities at every iteration of the optimisation, whereas our method avoids this extra cost. Not only is our method superior to these approaches in terms of cost functions (see Fig. 4), but it is computationally cheaper to evaluate as it does not require empirical distributions. This is a significant benefit because it not only reduces the cost per iteration, but avoids the issue of having to build 'good' distributions. One explanation for difference between the performance of these methods and ours, is that it is hard to build 'good' empirical distributions in real-time and most methods rely on simple histograms. Although this could be improved with Parzen or NP windowing techniques [19], it would almost certainly sacrifice real-time performance.

7.1 Timing

All terms in (17) include $\delta_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))$ (blurred Dirac delta function). This means that an individual pixel's contribution to the optimisation diminishes the further

from the contour it is. An efficient implementation, therefore, recognises this. Our implementation ignores pixels outside a narrow band and for an object size of 180×180 runs in 500μs on a P4 3.6GHz machine. On average the system runs at a frame rate of 85Hz for the complete algorithm and if shape and appearance learning are turned off (i.e. rigid registration only) it averages 230Hz.

8 Conclusions

We have proposed a novel probabilistic framework for robust, real-time, visual tracking of previously unseen objects from a moving camera. The key contribution of our method and reason for its superior performance compared with others is the use of pixel-wise posteriors as opposed to a product over pixel-wise likelihoods. In contrast to other methods [4,5], we solve the registration using Gauss Newton, which has significant practical benefits, namely: (i) the difficulty associated with step size selection is removed and (ii) reliable and fast convergence. We have demonstrated the benefits of our method both qualitatively and quantitatively with a thorough analysis of pixel-wise posteriors versus competing alternatives using over 20,000 video frames. Our results demonstrate that using pixel-wise posteriors provides excellent performance when incorporating non-parametric distributions into region based level-sets. It not only offers superior cost functions but avoids the need for computing empirical distributions [12,8,9,11] and is therefore faster.

In our ongoing research we are investigating in more detail the probabilistic interpretation of the blurred Heaviside step functions, and in particular the link with Gaussian uncertainty on contour location. Other work of future interest involves modifying the generative model to capture spatial information, and investigating how to incorporate multiple objects and online occlusion handling.

References

1. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, Secaucus, NJ, USA (2003)
2. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(3), 266–280 (2000)
3. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. IEEE Trans. on Image Processing 10(10), 1467–1475 (2001)
4. Cremers, D.: Dynamical statistical shape priors for level set based tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(8), 1262–1273 (2006)
5. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. International Journal of Computer Vision V72(2), 195–215 (2007)
6. Jebara, T.: Images as bags of pixels. In: Proc. 9th Int'l Conf. on Computer Vision, Nice (2003)
7. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing 10(2), 266–277 (2001)

8. Freedman, D., Zhang, T.: Active contours for tracking distributions. IEEE Transactions on Image Processing 13(4), 518–526 (2004)
9. Zhang, T., Freedman, D.: Improving performance of distribution tracking through background matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(2), 282–287 (2005)
10. Li, C., Xu, C., Gui, C., Fox, M.D.: Level set evolution without re-initialization: A new variational formulation. In: Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, California, vol. 1, pp. 430–436. IEEE Computer Society, Los Alamitos (2005)
11. Yilmaz, A.: Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, Minnesota (2007)
12. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, vol. 2, pp. 142–149 (2000)
13. Collins, R.T.: Mean-shift blob tracking through scale space. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 234–240 (2003)
14. Baker, S., Matthews, I.: Lukas-kanade 20 years on: A unifiying framework. International Journal of Computer Vision 69(3), 221–255 (2004)
15. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (SIGGRAPH 2004) (2004)
16. Evans, L.C.: Partial Differential Equations. AMS (2002)
17. Cremers, D., Osher, S.J., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. International Journal of Computer Vision 69(3), 335–351 (2006)
18. Fisher, R.: CAVIAR Test Case Scenarios, EC Funded IST 2001 37540. Online Book (October 2004)
19. Kadir, T., Brady, M.: Estimating statistics in arbitrary regions of interest. In: Proc. 16th British Machine Vision Conf., Oxford (2005)