Filtering Using a Tree-Based Estimator

B. Stenger * A. Thayananthan * P. H. S. Torr † R. Cipolla *

* University of Cambridge
Department of Engineering
Cambridge, CB2 1PZ, UK
{bdrs2|at315|cipolla}@eng.cam.ac.uk

† Microsoft Research Ltd.
7 JJ Thompson Avenue
Cambridge, CB3 OFB, UK
philtorr@microsoft.com

Abstract

Within this paper a new framework for Bayesian tracking is presented, which approximates the posterior distribution at multiple resolutions. We propose a tree-based representation of the distribution, where the leaves define a partition of the state space with piecewise constant density. The advantage of this representation is that regions with low probability mass can be rapidly discarded in a hierarchical search, and the distribution can be approximated to arbitrary precision. We demonstrate the effectiveness of the technique by using it for tracking 3D articulated and non-rigid motion in front of cluttered background. More specifically, we are interested in estimating the joint angles, position and orientation of a 3D hand model in order to drive an avatar.

1. Introduction

One of the fundamental problems in vision is that of tracking objects through sequences of images. Within this paper we present a generic Bayesian algorithm for tracking the 3D position and orientation of rigid or non-rigid objects (in our application hands) in monocular video sequences. Great strides have been made in the theory and practice of tracking, e.g. the development of particle filters recognized that a key aspect in tracking was a better representation of the posterior distribution of model parameters [**?**, **?**]. Particle filters go beyond the uni-modal Gaussian assumption of the Kalman filter by approximating arbitrary distributions with a set of random samples. The advantage is that the filter can deal with clutter and ambiguous situations more effectively, by not placing its bet on just one hypothesis. However, a major concern is that the number of particles required increases exponentially with the dimension of the state space [**?**, **?**]. Worse still, even for low dimensional spaces there is a tendency for particles to become concentrated in a single mode of the distribution [**?**]. Within this paper we consider tracking an articulated hand in cluttered images, without the use of markers, with the aim of driving an avatar. In general this motion has 27 degrees of freedom (DOF), 21 DOF for the joint angles and 6 for orientation and location [**?**]. However, by reparameterization the state space can be reduced. Wu *et al.* [**?**] show that due to the correlation of joint angles, the state space for the joints can be reduced to 7 DOF by applying PCA, with loss of only 5 percent of information, however tracking is demonstrated for a fixed view with no clutter and no hand rotation. We demonstrate 8 DOF tracking in clutter with substantial self-occlusion.

There are several possible strategies for estimation in high dimensional spaces. One way is to use a sequential search, in which some parameters are estimated first, and then others, assuming that the initial set of parameters is correctly estimated. This strategy may seem suitable for articulated objects. For example, Gavrila and Davis [**?**] suggest, in the context of human body tracking, first locating the torso and then using this information to search for the limbs. Unfortunately, this approach is in general not robust to different view points and self-occlusion. MacCormick and Isard [**?**] propose a particle filtering framework for this type of method in the context of hand tracking, factoring the posterior into a product of conditionally independent variables. This assumption is essentially the same as that of Gavrila and Davis, and tracking has been demonstrated only for a single view point with no self-occlusion.

The development of particle filters was primarily motivated by the need to overcome ambiguous frames in a video sequence so that the tracker is able to recover. Another way to overcome the problem of losing lock is to treat tracking as object detection at each frame. Thus if the target is lost in one frame, this does not affect any subsequent frame. Template based methods have yielded good results for locating deformable objects in a scene with no prior knowledge, e.g. for hands or pedestrians [**?**, **?**, **?**]. These methods are made robust and efficient by the use of distance transforms such as the chamfer or Hausdorff distance between template and image [**?**, **?**], and were originally developed for matching a single template. A key suggestion was that multiple templates could be dealt with efficiently by building a tree of

templates [**?**, **?**]. Given the success of these methods, it is natural to consider whether or not tracking might not be best effected by template matching using exhaustive search at each frame. The answer to this question is generally no, because dynamic information is needed, firstly to resolve ambiguous situations, and secondly, to smooth the motion. One approach to embed template matching in a probabilistic tracking framework was proposed by Toyama and Blake [**?**]. However, it is acknowledged that "one problem with exemplar sets is that they can grow exponentially with object complexity. Tree structures appear to be an effective way to deal with this problem, and we would like to find effective ways of using them in a probabilistic setting" [**?**]. Within this paper we address this problem.

The next section reviews work on tree-based detection, and describes how a tree can be used to partition a state space. A short review of Bayesian filtering is given in section 3. In section 4 we show how the tree-based partition of the state space can be embedded in a Bayesian filtering framework. The likelihood and state transition distributions for the application of hand tracking are derived in section 5. Section 6 shows tracking results on video sequences.

2. Tree-Based Detection

When matching many similar templates to an image, a significant speed-up can be achieved by forming a template hierarchy and using a coarse to fine search [**?**, **?**]. The idea is to group similar templates and represent them with a single prototype template together with an estimate of the variance of the error within the cluster, which is used to define a matching threshold. The prototype is first compared to the image; only if the error is below the threshold are the templates within the cluster compared to the image. This clustering is done at various levels, resulting in a hierarchy, with the templates at the leaf level covering the space of all possible templates. Gavrila [**?**] suggests forming the hierarchy by recursive (off-line) clustering, resulting in efficient on-line evaluation. When the exemplar templates are clustered using a cost function based on chamfer distance, the being not to miss objects when pruning sub-trees during the search. However, it is not straightforward how to give such guarantees when incorporating a prior for each template. In section 4 we show how a tree-based algorithm can be formulated in a Bayesian setting, using both likelihood and prior information.

If a parametric object model is available, another option to build the tree is by partitioning the state space. Let this tree have L levels, each level l defines a partition $\mathcal{P}_l$ of the state space into N_l distinct sets $l = 1, \ldots, L$, such that $\mathcal{P}_l = \{\mathcal{S}^{il} : i = 1, \ldots, N_l\}$. The leaves of the tree define the finest partition of the state space $\mathcal{P}_L = \{\mathcal{S}^{iL} : i = 1, \ldots, N_L\}$. Such a tree is depicted schematically in figure 1(a), for a single rotation parameter. This tree representation has the advantage that prior information is encoded efficiently, as templates with large distance in parameter space are likely to be in different sub-trees. In our particular case, a parametric three-dimensional hand model is used, shown in figure 3. The model has 6 DOF for rigid body motion and 21 DOF for finger articulation [**?**].

Detection as Optimal Estimation It is possible, after reaching the leaf level in a search tree, to use a gradient descent method to obtain the globally optimal parameters. This presents a trade-off between the number of function evaluations required for tree-based estimation and the number required for gradient descent, i.e. how many levels should there be in the tree before optimization is started? Furthermore we would like to guarantee that optimization, when started from one of the nodes at the leaf level, yields a global optimum. It may be argued that there is no need for a parametric model and that an exemplar-based approach could be followed. However, for models with many degrees of freedom the storage space for templates becomes excessive. The use of a parametric model allows the combination of an on-line and off-line approach in the tree-based algorithm. Once the leaf level is reached, it is possible that we are still not near to the global minimum, and further child templates can be generated.

Hierarchical detection works well for locating a hand in images [**?**] , and yet often there are ambiguous situations that could be resolved by using temporal information. The next section describes the Bayesian framework for filtering. Filtering is the problem of estimating the state (hidden variables) of a system given a history of observations.

3. Bayesian Filtering

Define, at time t, the state parameter vector as $\boldsymbol{\theta}_t$, and the data (observations) as $\mathbf{D}_t$, with $\mathbf{D}_{0:t-1}$, being the set of data from time 0 to $t-1$; and the data $\mathbf{D}_t$ are conditionally independent at each time step given the $\boldsymbol{\theta}_t$. In our specific application $\boldsymbol{\theta}_t$ is the state of the hand (set of joint angles, location and orientation) and $\mathbf{D}_t$ is the image at time t (or some set of features extracted from that image). Thus at time t the posterior distribution of the state vector is given by the following recursive relation

$$\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t}) = \frac{\Pr(\mathbf{D}_t|\boldsymbol{\theta}_t)\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t-1})}{\Pr(\mathbf{D}_t|\mathbf{D}_{0:t-1})}, \qquad (1)$$

where the normalizing constant is

$$\Pr(\mathbf{D}_t|\mathbf{D}_{0:t-1}) = \int \Pr(\mathbf{D}_t|\boldsymbol{\theta}_t)\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t-1})d\boldsymbol{\theta}_t. \quad (2)$$

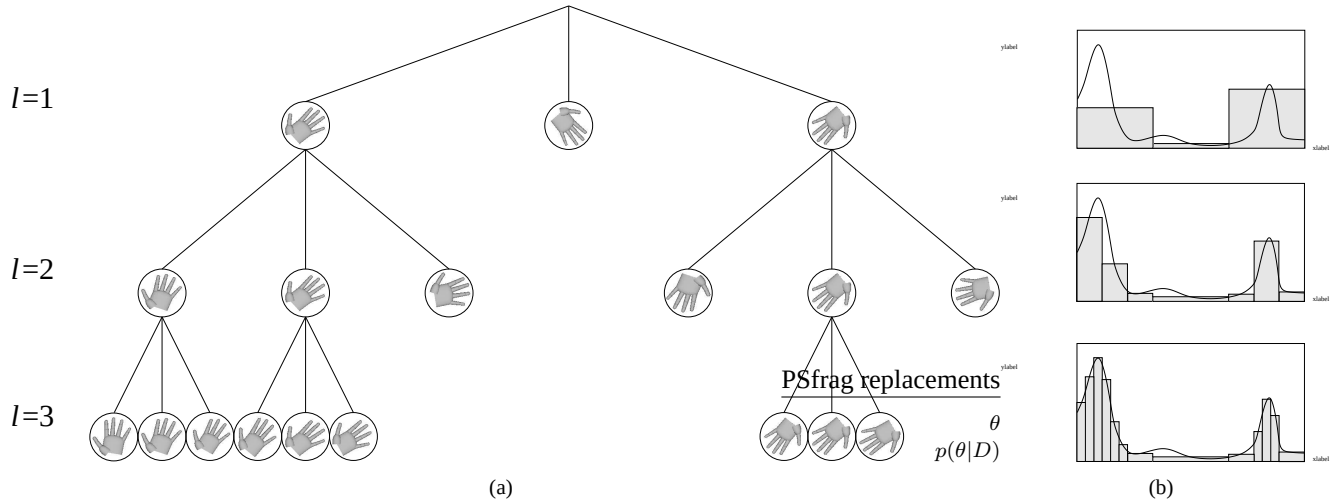


Figure 1: **Tree-based estimation of the posterior density.** *(a) Associated with the nodes at each level is a non-overlapping set in the state space, defining a partition of the state space (here rotation angle). The posterior for each node is evaluated using the center of each set, depicted by a hand rotated by a specific angle. Sub-trees of nodes with low posterior are not further evaluated. (b) Corresponding posterior density (continuous) and the piecewise constant approximation using tree-based estimation. The modes of the distribution are approximated with higher precision at each level.*

The term $\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t-1})$ in (1) is obtained from the Chapman-Kolmogorov equation:

$$\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t-1}) = \int \Pr(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) \Pr(\boldsymbol{\theta}_{t-1}|\mathbf{D}_{0:t-1}) d\boldsymbol{\theta}_{t-1} \tag{3}$$

with the initial prior distribution $\Pr(\boldsymbol{\theta}_0|\mathbf{D}_0)$ assumed known. It can be seen that (1) and (3) both involve integrals. Except for certain simple distributions these integrals are intractable and so approximation methods must be used. As has been mentioned, Monte Carlo methods represent one way of evaluating these integrals. However, as has been pointed out, there are many problems with particle filters in high dimensional spaces. In contrast hierarchical detection provides a very efficient way to sample the likelihood $\Pr(\mathbf{D}_t|\boldsymbol{\theta}_t)$ in a deterministic manner, even when the state space is high dimensional; as the number of templates in the tree increases exponentially with the number of levels in the tree. This leads us to consider the seminal approach of Bucy and Senne [**?**], which is to divide up the state space into N_s non-overlapping sets (a cover), $\{\mathcal{S}_t^i : i = 1, \ldots, N_s\}$, just as the templates in the tree cover the regions of parameter space. Typically this methodology has been applied using an evenly spaced grid and is thus exponentially expensive as the dimension of the state space increases e.g. [**?**]. Within this paper we consider combining the tracking process and the empirically successful process of tree-based detection as laid out in section 2 resulting in an efficient deterministic filter.

4. Tree-Based Filtering

Our aim is to design an algorithm that can take advantage of the efficiency of the tree-based search whilst also yielding a good approximation to Bayesian filtering. Sorenson [**?**] identifies three questions to be answered when designing a 'grid-based' filter, the questions (and our answers) are:

1. An initial partition must be defined on the state space. *In our case a natural multi-resolution partition is provided by the tree as given in Section 2. Thus we will consider a grid defined by the lowest leaves of the tree, $\mathcal{P}_L$.*

2. A procedure must be given for updating the partition as time progresses. *Because the distribution is characterized by being almost zero in large regions of the state space with some isolated peaks, many of the grid regions can be discarded as possessing negligible probability mass. The tree-based search provides an efficient way to rapidly concentrate computation on significant regions.*

3. Given the partition a method for approximating the distribution needs to be defined. *At the lowest level of the tree the distribution will be assumed to be piecewise constant, which will be seen to allow for some reasonable approximations to be made to the Bayesian filtering equations.*

The plan is to encode the posterior distribution using a piecewise constant distribution over the leaves of the tree.

This distribution will be mostly zero for many of the leaves. To formalize this as a discrete problem, define Θ_t^{il} as the parameter values in the state space integrated over the region $\mathcal{S}^{il}$ at time t, i.e.

$$\Pr(\Theta_t^{il}) = \int_{\boldsymbol{\theta}_t \in \mathcal{S}^{il}} \Pr(\boldsymbol{\theta}_t) d\boldsymbol{\theta}_t. \tag{4}$$

For each layer of the tree we consider the distribution over the $\mathcal{S}^{il}$ and recast the equations of Bayesian filtering, (1)-(3), to update these distributions. The initial prior distribution for the discrete states $\Pr(\Theta_0^{iL}|\mathbf{D}_0)$ can be obtained by integration from $\Pr(\boldsymbol{\theta}_0|\mathbf{D}_0)$, as

$$\Pr(\Theta_0^{iL}|\mathbf{D}_0) = \int_{\boldsymbol{\theta}_0 \in \mathcal{S}^{iL}} \Pr(\boldsymbol{\theta}_0|\mathbf{D}_0) d\boldsymbol{\theta}_0. \tag{5}$$

Next the discrete recursive relations are defined, again these are obtained from the continuous case by integration.

Given the distribution over the leaves of the tree, $\Pr(\Theta_{t-1}^{iL}|\mathbf{D}_{0:t-1})$, at the previous time step $t-1$, equation (3) now becomes a transition between discrete regions in state space:

$$\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1}) = \sum_{i=1}^{N_L} \Pr(\Theta_t^{jl}|\Theta_{t-1}^{iL}) \ \Pr(\Theta_{t-1}^{iL}|\mathbf{D}_{0:t-1}). \tag{6}$$

Assuming the conditional distribution, $\Pr(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$, is known, then

$$\Pr(\Theta_t^{jl}|\Theta_{t-1}^{iL}) = \int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} \int_{\boldsymbol{\theta}_{t-1} \in \mathcal{S}^{iL}} \Pr(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) d\boldsymbol{\theta}_t d\boldsymbol{\theta}_{t-1}. \tag{7}$$

Although this is somewhat intractable, it can be approximated using numerical integration methods and stored in a look up table ahead of time. (An alternative approach is to acquire large amounts of training data and learn the state transition probabilities.) This is not the case for the posterior $\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t})$. Given that the distribution of $\boldsymbol{\theta}_t$ is piecewise constant within each $\mathcal{S}^{jl}$, then for $\boldsymbol{\theta}_t \in \mathcal{S}^{jl}$: $\Pr(\boldsymbol{\theta}_t|\mathbf{D}_{0:t-1}) = \Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1})/\gamma^{jl}$, where γ^{jl} is the volume of $\mathcal{S}^{jl}$. With this key assumption the posterior (1) becomes

$$\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t}) = \frac{\Pr(\mathbf{D}_t|\Theta_t^{jl}) \ \Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1})}{\Pr(\mathbf{D}_t|\mathbf{D}_{0:t-1})\,\gamma^{jl}}, \tag{8}$$

where

$$\Pr(\mathbf{D}_t|\Theta_t^{jl}) = \int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} \Pr(\mathbf{D}_t|\boldsymbol{\theta}_t)\, d\boldsymbol{\theta}_t. \tag{9}$$

The normalization constant is

$$\Pr(\mathbf{D}_t|\mathbf{D}_{0:t-1}) = \sum_{j=1}^{N_l} \Pr(\mathbf{D}_t|\Theta_t^{jl}) \ \frac{\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1})}{\gamma^{jl}}. \tag{10}$$

The likelihood in (8) and normalizing constant (10) cannot be computed off-line as they depend on the data, $\mathbf{D}_t$, at time t. The integral in (9) is often intractable hence the approach we adopt is to approximate it by using the rectangle rule or Riemann sum with one subdivision per set $\mathcal{S}^{jl}$, based on the height (likelihood) estimated at $\boldsymbol{\theta}^{jl}$, the center of $\mathcal{S}^{jl}$:

$$\Pr(\mathbf{D}_t|\Theta_t^{jl}) \ \approx \ \gamma^{jl} \Pr(\mathbf{D}_t|\boldsymbol{\theta}^{jl}). \tag{11}$$

As the number of partitions increases this becomes an increasingly close approximation to the true distribution.

Having laid out Bayesian filtering over discrete states the question arises how do we combine the theory with the efficient tree-based algorithm previously described. Using a breadth first search of the tree, the posterior may be approximated by using (6)-(11) at each each level. At each level the regions with high posterior are identified and explored in finer detail in the next level (Figure 1b). Of course it is to be expected that the higher levels will not yield accurate approximations to the posterior. However, just as for the case of detection, the upper levels of the tree are used to discard inadequate hypotheses, for which the negative log posterior of the set exceeds a threshold (which is adapted to the level of the tree), and verily efficiency is assured. The thresholds at the higher levels of the tree are set conservatively so as to not discard good hypotheses too soon. An overview of the algorithm is given in Algorithm 1.

Algorithm 1 Tree-Based Filtering

1. Initialization, $t = 0$

At the first level of the tree, $l = 1$

$\Pr(\Theta_0^{j1}|\mathbf{D}_0) = \frac{1}{K} \Pr(\mathbf{D}_0|\Theta_0^{j1})$

At higher levels of the tree, $l > 1$

$$\Pr(\Theta_0^{jl}|\mathbf{D}_0) = \begin{cases} \frac{1}{K} \Pr(\mathbf{D}_0|\Theta_0^{jl}) & \text{if } \Pr(\Theta_0^{(k)(l-1)}|\mathbf{D}_0) > \rho_{l-1} \\ \Pr(\Theta_0^{(k)(l-1)}|\mathbf{D}_0) & \text{otherwise} \end{cases}$$

2. At time $t > 0$

At the first level of the tree, $l = 1$

$\Pr(\Theta_t^{j1}|\mathbf{D}_{0:t}) = \frac{1}{K} \Pr(\mathbf{D}_t|\Theta_t^{j1}) \ \Pr(\Theta_t^{j1}|\mathbf{D}_{0:t-1})$

At higher levels of the tree, $l > 1$

$$\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t}) = \begin{cases} \frac{1}{K} \Pr(\mathbf{D}_t|\Theta_t^{jl}) \ \Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1}) & \\ \qquad \text{if } \Pr(\Theta_t^{(k)(l-1)}|\mathbf{D}_{0:t}) > \rho_{l-1} & \\ \Pr(\Theta_t^{(k)(l-1)}|\mathbf{D}_{0:t}) \ \text{otherwise} & \end{cases}$$

where

$\Pr(\Theta_t^{jl}|\mathbf{D}_{0:t-1}) = \sum_{i=1}^{N_L} \Pr(\Theta_t^{jl}|\Theta_{t-1}^{iL}) \ \Pr(\Theta_{t-1}^{iL}|\mathbf{D}_{0:t-1})$

K is the normalization constant in each equation

k is the parent node at the previous level of the tree

ρ_l is the threshold value at level l of the tree

5. Formulating Likelihood and Transition Distribution

This section explains the likelihood and state transition distribution which are used for tracking a hand.

5.1. Formulating the Likelihood

The likelihood function relates the observations $\mathbf{D}_t$ to the unknown state $\boldsymbol{\theta}_t$. For hand tracking, color and edges features have been used frequently in the past [**?**, **?**, **?**, **?**]. Thus the data is taken to be composed of two sets of observations, those from edge data $\mathbf{D}_t^{edge}$ and from color data $\mathbf{D}_t^{col}$. The likelihood function used is

$$\log p(\mathbf{D}_t|\boldsymbol{\theta}_t) = \log p(\mathbf{D}_t^{edge}|\boldsymbol{\theta}_t) + \lambda \log p(\mathbf{D}_t^{col}|\boldsymbol{\theta}_t), \tag{12}$$

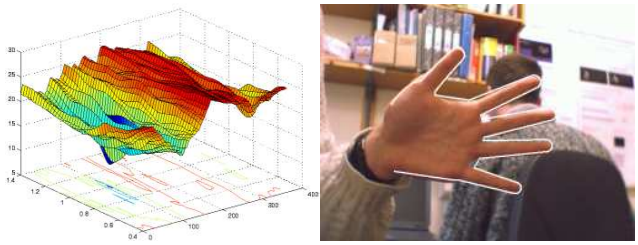
where λ is a weighting parameter. The term for edge contours, $p(\mathbf{D}_t^{edge}|\boldsymbol{\theta}_t)$, is based on the chamfer distance function [**?**, **?**]. Given the set of projected model contour points, $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^n$, and the set of Canny edge points, $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^m$, a quadratic chamfer distance function is given by

$$d^2_{cham}(\mathcal{U}, \mathcal{V}) = \frac{1}{n}\sum_{i=1}^{n} d^2(i, \mathcal{V}), \tag{13}$$

where $d(i, \mathcal{V}) = \max(\min_{v_j \in \mathcal{V}} ||u_i - v_j||, \tau)$ is the thresholded distance between the point, $u_i \in \mathcal{U}$, and its closest point in $\mathcal{V}$. Using a threshold value τ makes the matching more robust to outliers and missing edges. The chamfer distance between two shapes can be computed efficiently using a distance transform, where the template edge points are correlated with the distance transform of the image edge map. Edge orientation is included by computing the distance only for edges with similar orientation, in order to make the distance function more robust [**?**]. We also exploit the fact that part of an edge normal on the interior of the contour should be skin-colored, and only take those edges into account [**?**]. In constructing the color likelihood function $p(\mathbf{D}_t^{col}|\boldsymbol{\theta}_t)$, we seek to explain all the image pixel data given the proposed state. Given a state, the pixels in the image $\mathcal{I}$ are partitioned into a set of object pixels $\mathcal{O}$, and a set of background pixels $\mathcal{B}$. Assuming pixel-wise independence, the likelihood can be factored as

$$p(\mathbf{D}_t^{col}|\boldsymbol{\theta}_t) = \prod_{o \in \mathcal{O}} p(I_t(o)|\boldsymbol{\theta}_t) \prod_{b \in \mathcal{B}} p(I_t(b)|\boldsymbol{\theta}_t), \tag{14}$$

where $I_t(k)$ is the intensity normalized rg-color vector at pixel location k at time t. The object color distribution is modeled as a Gaussian distribution in the normalized color space [**?**], for background pixels a uniform distribution is assumed. For efficiency, we evaluate only the edge likelihood term while traversing the tree, and incorporate the color likelihood only at the leaf level.



(a) (b)

Figure 2: **Negative Log-Likelihood surface with single global minimum**. *(a) Surface described by the negative log-likelihood function when searching the scale and angle space, matching a template with the input image shown in (b). The superimposed template corresponds to the global minimum in (a), but there are many local minima.*

Figure 2 shows a plot of the negative log-likelihood surface, generated by varying two parameters, angle and scale, around the best matching model for a particular image. The global minimum is at the correct location, but there are many local minima.

5.2. Formulating the Transition Distribution

Natural hand articulation is constrained, and Wu *et al.* [**?**] have shown that the vectors of valid joint angles lie on a lower dimensional manifold, which is approximated as the union of linear manifolds. We use data from marker-based motion capture experiments to obtain points in the state space. Figure 3 shows the projection of points into the space of the three joint angles of the index finger during its flexion and extension. This non-linear manifold is parameterized by approximating it with a piecewise linear function. This parameterization is used to generate templates for articulated motion, corresponding to a valid set of joint angle values. The state transition distribution is assumed to be Gaussian

$$p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) \sim \mathcal{N}(\boldsymbol{\theta}_{t-1}, \Sigma), \tag{15}$$

where Σ is a diagonal covariance matrix. This is a simplified model, but a dynamical model for hand motion, learned from training data, can be integrated in this step [**?**].

One of the advantages of using a parametric 3D model is that the transition probabilities have an intrinsic physical meaning, e.g. the rate of change of a joint angle. This is in contrast to 2D shape-based methods which require a large amount of training data before they can be fully specified.

6. Results

We demonstrate the effectiveness of our technique by tracking both hand motion and finger articulation in cluttered

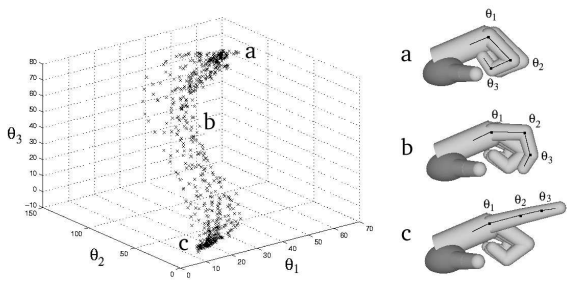


Figure 3: **Manifold in state space described by joint angles**. *Three joint angles of index finger during flexion and extension. Each data point corresponds to one particular pose of the finger.*

scenes using a single camera. The results reveal the ability of the tree-based filter to handle ambiguity arising from self-occlusion and 3D motion.

6.1. 3D Hand Tracking Experiments

In the hand tracking experiments templates are generated by projecting a 3D hand model described in [**?**]. In two video sequences we track the global 3D motion of the hand without finger articulation. The 3D rotations are limited to a hemisphere. A three-level tree is built which has the following resolutions at the leaf level: 15 degrees in two 3D rotations, 10 degrees in image rotation and 5 different scales, resulting in a total of $13 \times 13 \times 19 \times 5 = 16055$ templates. The resolution of the translation parameters is 20 pixels on the first level, 5 pixels on the second level, and 2 pixels on the leaf level. Figures 4 and 5 show results from tracking a pointing and an open hand, respectively, through their global motions.

In the third sequence (figure 6) tracking is demonstrated for global hand motion together with finger articulation. A piecewise linear approximation to the manifolds described in section 5.2 is used to model finger articulation. The articulation parameters for the thumb and fingers are approximated using 7 and 5 subdivisions in the valid range, respectively. For this sequence the range of global hand motion is restricted to a smaller region, but it still has 6 DOF. In total 35000 templates are used at the leaf level. The resolution of the translation parameters is the same as in the first experiments.

The computation takes approximately two seconds per frame on a 1GHz Pentium IV. Note that in all three cases, the hand model was automatically initialized by searching the tree in the first frame of the sequence.

7. Summary and Conclusion

This paper endeavors to narrow the gap between detection and tracking, in order to enjoy the benefits of both worlds. Reliable detection helps in dealing with difficult problems such as self-occlusion. Tracking embeds detection in a filtering framework, making use of dynamic information. It also makes detection more efficient by eliminating a significant number of hypotheses.

To make this marriage, we cast the problem in a probabilistic framework. Bayesian methods are attractive as they provide a principled way of encoding uncertainty and multiple hypotheses about parameter estimates. This is particularly necessary for the problem of tracking in clutter as there is much ambiguity, resulting in multi-modal distributions. One of the key issues in Bayesian filtering is how to represent these distributions. Previously grid-based methods, involving partitioning the state space, have proven very successful for propagating distributions in tracking. However, they suffer from the major draw back that they are computationally infeasible in high dimensional spaces. In order to cope with this we propose a tree-based representation which can be used to select grid points (leaves or partitions of the state space) with high probability mass to represent the distribution.

We have tested the new tracking method on sequences involving clutter in the background together with non-rigid hand motion. Furthermore within these sequences the hand undergoes large rotations leading to significant topological changes in the projected contours. The tracker produces very good results even in these circumstances. Finally we observe that the method of partitioning the state space and using a tree-based search to propagate distributions is a generic method that can be applied to other tracking problems.

Acknowledgments The authors would like to thank Michael Isard and Frank Dellaert for insightful discussions. This work was supported in part by the Gottlieb Daimler– and Karl Benz–Foundation, the EPSRC, the Gates Cambridge Trust, and the Overseas Research Scholarship Programme.

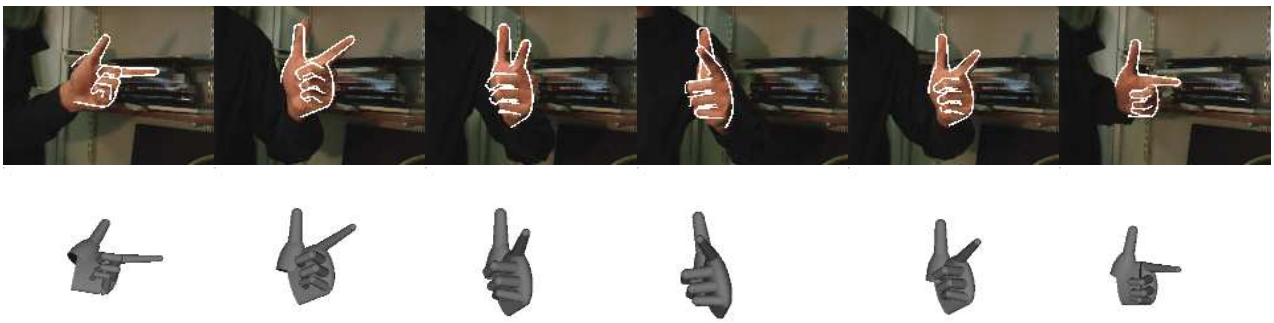


Figure 4: **Tracking a pointing hand in front of clutter**. *The images are shown with projected contours superimposed (top) and corresponding 3D avatar models (bottom), which are estimated using the tree-based filter. The hand is translating and rotating. A 2D deformable template would have problems coping with topological shape changes caused by self-occlusion.*

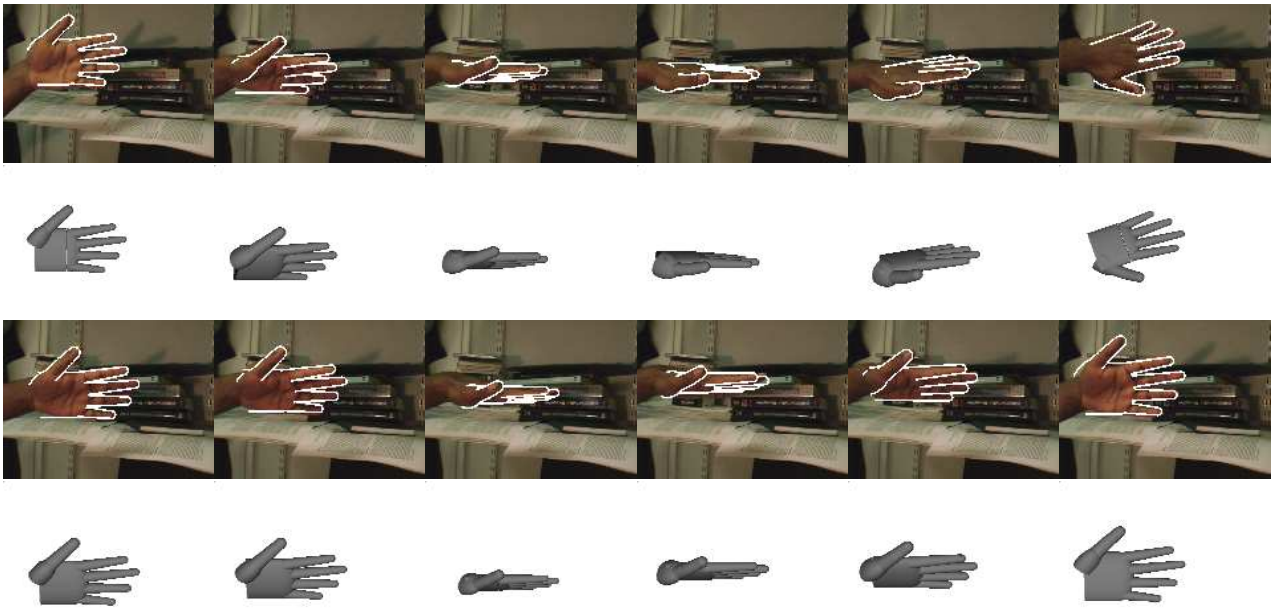


Figure 5: **Tracking a flat hand rotating in clutter.** *In this sequence the hand undergoes rotation and translation. The frames showing the hand with significant self-occlusion do not provide much data, and template matching becomes unreliable. By including prior information, these situations can be resolved. The projected contours are superimposed on the images, and the corresponding 3D avatar model, which is estimated using the tree-based filter, is shown below each frame.*

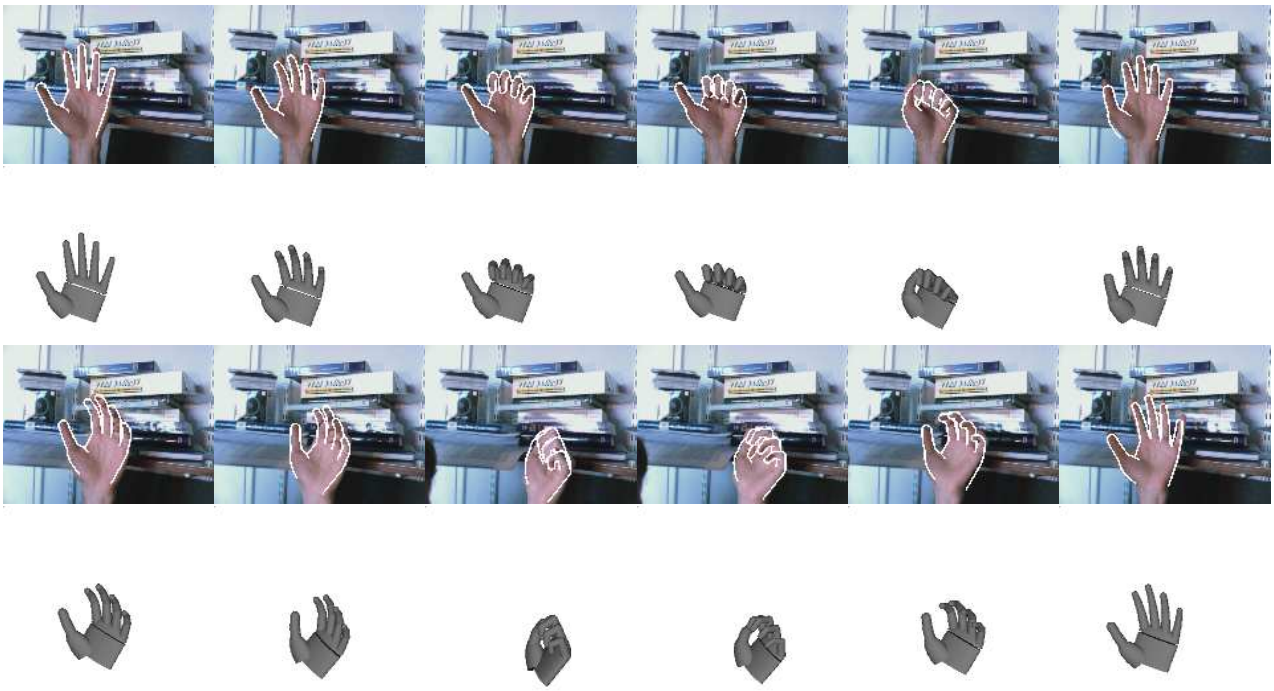


Figure 6: **Tracking a hand opening and closing with rigid body motion in front of a cluttered background.** *This sequence is challenging because the hand undergoes translation and rotation while opening and closing the fingers. 6* DOF *for rigid body motion plus 2* DOF *using manifolds for finger flexion and extension are tracked successfully.*