Filtering Using a Tree-based Estimator

B. Stenger A. Thayananthan, P. H. S. Torr, R. Cipolla

CUED/F-INFENG/TR 456

26 May 2003

Department of Engineering
University of Cambridge
Trumpington Street
Cambridge CB2 1PZ, UK

{bdrs2|at315|cipolla}@eng.cam.ac.uk

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 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. Large sets of training data are captured using a data glove, and different techniques for constructing the tree from this data are suggested.

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 recognised that a key aspect in tracking was a better representation of the posterior distribution of model parameters [12, 14]. 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 [8, 16]. Worse still, even for low dimensional spaces there is a tendency for particles to become concentrated in a single mode of the distribution [9], which has to be dealt with by careful choice of an importance density and the use of resampling.

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 reparameterisation the state space can be reduced. Wu *et al.* [29] 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.

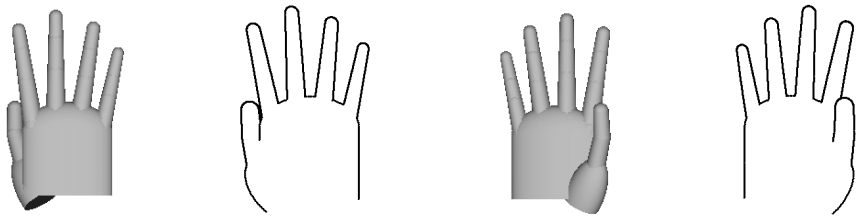
There are several possible strategies for estimation in high dimensional spaces. One way is to use a sequential search, in which some parameters are estimated first, and then others, assuming that the initial set of parameters is correctly estimated. This strategy may seem suitable for articulated objects. For example, Gavrila and Davis [11] 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 [16] 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 [2, 10, 21]. These methods are made robust and efficient by the use of distance transforms such as the chamfer or Hausdorff distance between template and image [4, 13], 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 [10, 18]. Given the success of these methods, it is natural to consider whether or not tracking might be best effected by template matching using exhaustive search at each frame. The answer to this question is no in general, 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 [27]. 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" [27]. 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 introduction to 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 function for hand tracking is derived in section 5, and the modelling of hand kinematics is explained in section 6. Section 7 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 [10, 18]. The idea is to



(a) (b)

Figure 1: **Poses of the hand with large parameter distance yet similar shape**. *Two poses of an open hand, which have large distance in the rotation space (180 degrees) but project to similar shapes. Clustering based on shape similarity may group them together, whereas partitioning of the state space will not, allowing different motion priors.*

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 [10] suggests forming the hierarchy by recursive (off-line) clustering, the goal being efficient on-line evaluation. When exemplar templates are clustered using a cost function based on chamfer distance, templates which look similar are likely to be in the same sub-tree. However, it is not straightforward to incorporate a prior for each template. For example, consider the case in figure 1, where two hand poses which are far apart in parameter space yield two similar templates. These templates may be clustered together, even though they have different motion priors. When building the tree in parameter space, however, the two configurations are very likely to be in different sub-trees, allowing for different priors. The matching threshold at each node should still be chosen according to the variation of an appearance similarity measure in the sub-tree.

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 2(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 1. The model has 6 DOF for rigid body motion and 21 DOF for finger articulation [24].

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 an interesting trade-off between the number of function evaluations required

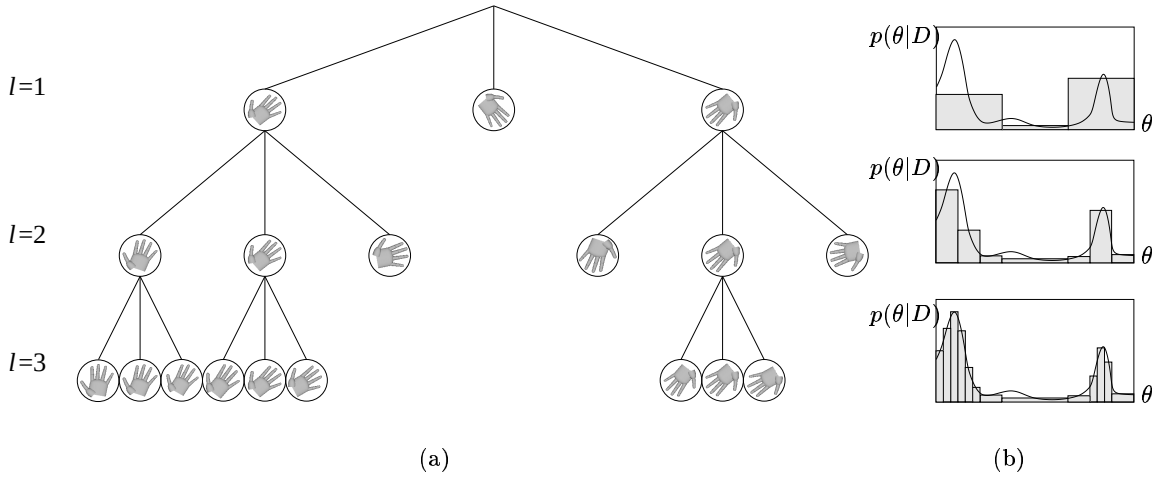


Figure 2: **Tree-based estimation of the posterior density.** *(a) Associated with the nodes at each level is a non-overlapping set in the state space, defining a partition of the state space (here rotation angle). The posterior pdf for each node is evaluated using the centre of each set, depicted by a hand rotated by a specific angle. Sub-trees of nodes with low posterior probability 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.*

for tree-based estimation and the number required for gradient descent, i.e. how many levels should there be in the tree before optimisation is started? Furthermore we would like to guarantee that optimisation, 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. However, 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 [26] , 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 $\mathbf{D}_{1:t-1}$, being the set of data from time 1 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

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

where the normalising constant is

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

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

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

with the initial prior pdf $p(\boldsymbol{\theta}_0|\mathbf{D}_0)$ 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 $p(\mathbf{D}_t|\boldsymbol{\theta}_t)$ 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 [7], 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. [5]. Within this paper we consider marrying the tracking process to the empirically successful process of tree-based detection as laid in Section 2 resulting in an efficient deterministic filter.

4 Filtering with a Tree-Based Estimator

The aim 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 [23] 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 characterised 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 pdf needs to be defined. *At the lowest level of the tree the pdf 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 formalise this as a discrete problem, define the set of states $\mathcal{S}_t^{il}$, where the $\mathcal{S}_t^{il}$ correspond to regions of state space at time t: $\{\boldsymbol{\theta}_t \in \mathcal{S}^{il}\}$. For each layer of the tree we consider the distribution over the $\mathcal{S}_t^{il}$ and recast the equations of Bayesian filtering, (1)-(3), to update these states.

The initial prior pdf for the discrete states $p(\mathcal{S}_0^{iL}|\mathbf{D}_0)$ can be obtained by integration from $p(\boldsymbol{\theta}_0|\mathbf{D}_0)$, as

$$p(\mathcal{S}_0^{iL}|\mathbf{D}_0) = \int_{\boldsymbol{\theta}_0 \in \mathcal{S}^{iL}} p(\boldsymbol{\theta}_0|\mathbf{D}_0) d\boldsymbol{\theta}_0. \tag{4}$$

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, $p(\mathcal{S}_{t-1}^{iL}|\mathbf{D}_{1:t-1})$, at the previous time step $t-1$. The Chapman-Kolmogorov equation (3) now becomes a transition between discrete states:

$$p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t-1}) = \sum_{i=1}^{N_L} p(\mathcal{S}_t^{jl}|\mathcal{S}_{t-1}^{iL})\ p(\mathcal{S}_{t-1}^{iL}|\mathbf{D}_{1:t-1}). \tag{5}$$

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

$$p(\mathcal{S}_t^{jl}|\mathcal{S}_{t-1}^{iL}) = \int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} \int_{\boldsymbol{\theta}_{t-1} \in \mathcal{S}^{iL}} p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) d\boldsymbol{\theta}_t d\boldsymbol{\theta}_{t-1}\ . \tag{6}$$

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 would be to acquire large amounts of training data and learn the state transition probabilities.) This is not the case for the posterior $p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t})$. Given that the distribution of $\boldsymbol{\theta}_t$ is piecewise constant within each $\mathcal{S}_t^{jl}$, then for $\boldsymbol{\theta}_t \in \mathcal{S}_t^{jl}$: $p(\boldsymbol{\theta}_t|\mathbf{D}_{1:t-1}) = p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t-1})/\gamma^{jl}$, where γ^{jl} is the volume of $\mathcal{S}^{jl}$. Then the posterior (1) becomes

$$p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t}) = \frac{\int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} p(\mathbf{D}_t|\boldsymbol{\theta}_t)\, d\boldsymbol{\theta}_t\ \ p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t-1})}{p(\mathbf{D}_t|\mathbf{D}_{1:t-1})\gamma^{jl}} \tag{7}$$

where the normalisation constant is

$$p(\mathbf{D}_t|\mathbf{D}_{1:t-1}) = \sum_{j=1}^{N_l} \int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} p(\mathbf{D}_t|\boldsymbol{\theta}_t) d\boldsymbol{\theta}_t \frac{p(\mathcal{S}_t^{jl}|\mathbf{D}_{1:t-1})}{\gamma^{jl}}. \tag{8}$$

The likelihood in (7) and normalising constant (8) cannot be computed off-line as they depend on the data, $\mathbf{D}_t$, at time t. The integral 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 $\mathcal{S}_t^{jl}$, based on the height (likelihood) estimated at the template, $\boldsymbol{\theta}_t^{jl}$, at the centre of $\mathcal{S}_t^{jl}$:

$$\int_{\boldsymbol{\theta}_t \in \mathcal{S}^{jl}} p(\mathbf{D}_t|\boldsymbol{\theta}_t) d\boldsymbol{\theta}_t \;\approx\; \gamma^{jl} p(\mathbf{D}_t|\boldsymbol{\theta}_t^{jl}). \tag{9}$$

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 to 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 (5)-(9) at each each level. At each level the regions with high posterior are identified and explored in finer detail in the next level (Figure 2b). 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 just 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. As in the case of tree-based detection, the threshold for each partition (tree node) should depend on the possible variation of the matching costs for candidates within that partition.

5 The Likelihood Function

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

5.1 Formulating the Likelihood

A key ingredient for any tracker is the likelihood function $p(\mathbf{D}_t|\boldsymbol{\theta}_t)$, which relates the observations $\mathbf{D}_t$ to the unknown state $\boldsymbol{\theta}_t$. Ideally the chosen observations should yield a likelihood with high discriminative power for detecting a hand, with as few local minima as possible. Furthermore it should be possible to compute the features and the likelihood with little computational overhead. For hand tracking, finding good features and a likelihood function is challenging, since there are few features which can be detected and tracked reliably. Colour values and edges contours appear to be suitable and have been used frequently in the past [2, 16, 29]. Thus the data is taken to be composed of two sets of observations, those from edge data $\mathbf{D}_t^{edge}$ and from colour data $\mathbf{D}_t^{col}$. The likelihood function is assumed to factor as

$$p(\mathbf{D}_t|\boldsymbol{\theta}_t) = p(\mathbf{D}_t^{edge}|\boldsymbol{\theta}_t)\; p(\mathbf{D}_t^{col}|\boldsymbol{\theta}_t). \tag{10}$$

The likelihood term for edge contours $p(\mathbf{D}_t^{edge}|\boldsymbol{\theta}_t)$ is based on the chamfer distance function [4, 6]. 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

$$d_{cham}^2(\mathcal{U}, \mathcal{V}) = \frac{1}{n} \sum_{i=1}^{n} d^2(i, \mathcal{V}), \tag{11}$$

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. Toyama and Blake [27] show how the quadratic chamfer function can be turned into a likelihood which is approximately Gaussian.

Edge orientation is included by computing the distance only for edges with similar orientation, in order to make the distance function more robust [18]. We also exploit the fact that part of an edge normal on the interior of the contour should be skin-coloured, and only take those edges into account [16].

In constructing the colour 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

$$p(\mathbf{D}_t^{col}|\boldsymbol{\theta}_t) = \prod_{k \in \mathcal{I}} p(I_t(k)|\boldsymbol{\theta}_t) \tag{12}$$

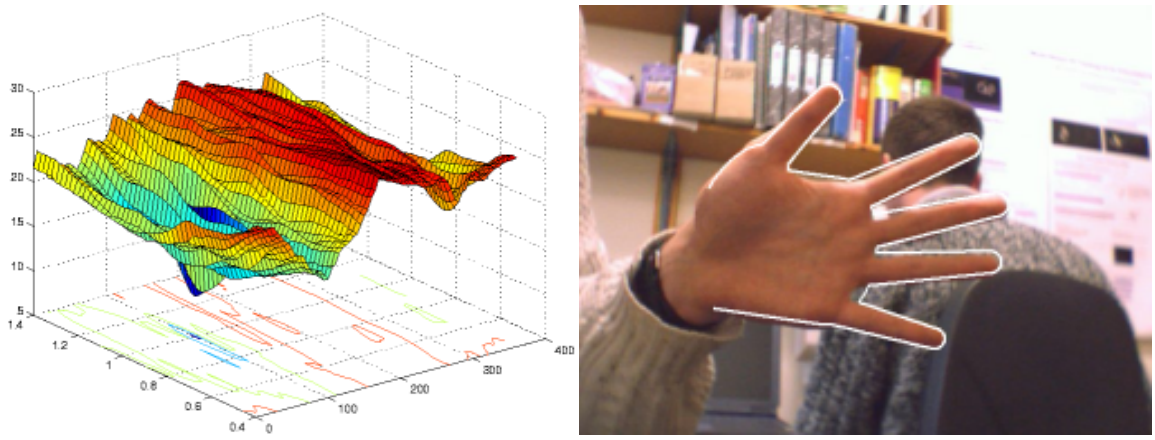
$$= \prod_{o \in \mathcal{O}} p(I_t(o)|\boldsymbol{\theta}_t) \prod_{b \in \mathcal{B}} p(I_t(b)|\boldsymbol{\theta}_t), \tag{13}$$

where $I_t(k)$ is the intensity normalised rg-colour vector $(\frac{R}{R+G+B}, \frac{G}{R+G+B})^T$ at pixel location k at time t. The object colour distribution is modeled as a Gaussian distribution in the normalised colour space [30]. The advantages of this representation are that skin colour values cluster in a small region in this colour space, and that the distribution parameters can be adapted efficiently. For background pixels a uniform distribution is assumed. However, improvements can be expected by learning a background colour model, e.g. a Gaussian mixture model. For efficiency, we evaluate only the edge likelihood term while traversing the tree, and incorporate the colour likelihood only at the leaf level.

Figure 3 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.

6 Modelling Hand Kinematics

Model-based trackers commonly use a 3D geometric model with an underlying biomechanical deformation model to represent the hand [1, 3, 20]. Each finger can be modelled as a kinematic chain with 4 degrees of freedom (DOF), and the thumb with 5 DOF. Thus articulated hand motion lies in a 21 dimensional joint angle space. Given a 3D hand model, inverse kinematics may be used to calculate the joint angles [28], however this problem is ill-posed when using a single view and it requires exact feature localisation, which is particularly difficult in the case of self-occlusion. However, hand motion is highly constrained as each joint can only move within certain limits. Furthermore the motion of different joints is correlated, for example, most people find it difficult to bend the little finger while keeping the ring finger fully extended at the same time. Thus hand articulation is expected to lie in a compact region within the 21 dimensional angle space. Wu *et al.* [29] represent articulated hand motion in a 7D configuration space. Joint angle measurements are obtained using a



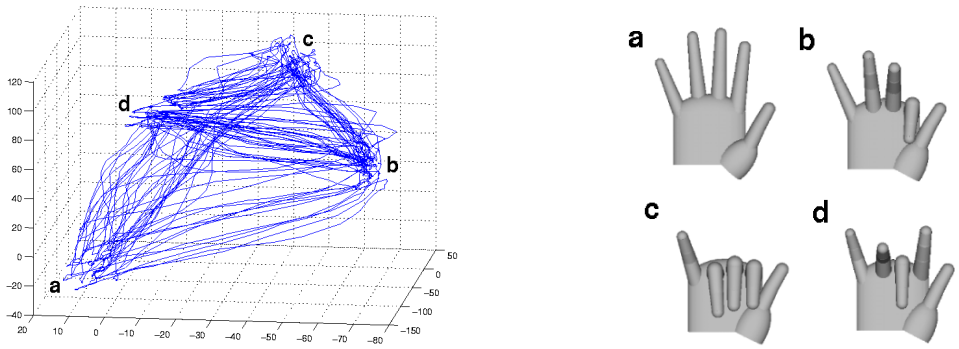
(a) (b)

Figure 3: **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.*

data glove, and they observe that the data can be projected into a 7D subspace using PCA with loss of only 5 percent of information. In this space 28 basis configurations are defined, corresponding to each finger being either fully extended or fully bent. It is claimed that natural hand articulation lies largely on linear 1D manifolds spanned by any two such basis configurations.

In a number of experiments with 15 sets of joint angles (sizes of data sets: 3,000 to 264,000) captured from three different subjects, we found in all cases that 95 percent of the variance was captured by the first eight principal components, in 10 cases within the first seven, which confirms the results reported in [29]. However, except for very controlled opening and closing of fingers, we did not find the one-dimensional linear manifold structure reported in [29]. For example, figure 4 shows trajectories (projected onto the first three eigenvectors) between a subset of "basis configurations" used in [29]. It can be seen that even in this controlled experiment the trajectories between the four configurations do not seem to be adequately represented by 1D manifolds.

In order to build the tree described in section 4, the state space needs to be partitioned at multiple resolutions. One way to do this is to cluster the data using a hierarchical k-means algorithm. This is computationally expensive for large data sets, but can be done off-line. The cluster centres in each level of the hierarchy are used as nodes in one level of the tree. A partition of the space is given by the Voronoi diagram defined by these nodes, see figure 5a. Another way to subdivide the space is to use a regular grid. Defining hard partition boundaries in the original 21 dimensional space is difficult though, because the number of partitions increases exponentially with the number of divisions along each axis. Therefore the data can first be projected onto the first k principal components ($k < 21$), and the partitioning is done in the transformed parameter space. The centres of these



(a) (b)

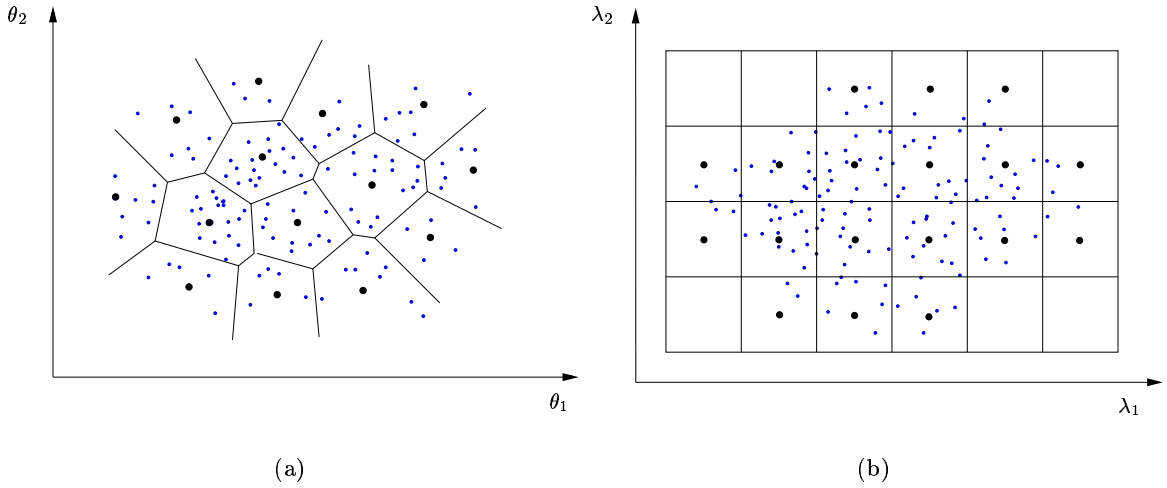
Figure 4: **Paths in the configuration space found by PCA.** *(a) The figure shows a trajectory of projected hand state vectors onto the first three principal components. (b) The hand configurations corresponding to the four end points in (a).*

hyper-cubes are then used as nodes in the tree on one level, where only partitions need to be considered which contain data points, see figure 5b. Multiple resolutions are obtained by subdividing each partition.

Given the partition of the space, the state transition distributions $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ can be modelled as a Markov process. We model it as a first order process, and compute the transition probabilities by histogramming transitions in the training set. Given a large amount of training data, higher order models can be learned.

Nonlinear Dimensionality Reduction In many cases the trajectories in eigenspace look rather nonlinear. Thus techniques for nonlinear dimensionality reduction, such as the Isomap [25] algorithm, may turn out to be more suitable to analyse the data. The Isomap algorithm is an extension of multidimensional scaling (MDS), In MDS the objective is to maintain pairwise distances between data points while reducing the dimensionality. Isomap uses approximate geodesic distances (trajectories on the manifold), computed from a locally connected graph, instead of Euclidean distances between data points. We have applied the Isomap algorithm to a number of data sets, but the first experiments show that the results are very dependent on choosing the right local neighbourhood to compute the pairwise distances. This is difficult, because some parts of the joint angle space are more densely sampled than others.

Comparison with Human Body Tracking Tracking articulated objects is often done in the context of human body tracking for which there exists a large amount of literature, see [17] for a survey. Some body trackers model the dynamics explicitly, for example using a switching linear dynamic system [19]. It might be possible to learn such a model for hand motion, but one main difference to full body motion is that there are no obvious motion patterns such as walking or running. (This may be the case, though, when considering only a set of well defined gestures). Another approach is to learn explicit probabilistic models such as hidden Markov models. For example, Karaulova *et al.* [15] do this in a



(a) (b)

Figure 5: **Partitioning the State Space.** *This figure illustrates two methods of partitioning the state space: (a) by clustering the data points in the original space, and (b) by first projecting the data points onto the first k principal components and then using a regular grid to define a partition in the transformed space. The figure only shows one level of a hierarchical partition, corresponding to one level in the tree.*

lower dimensional eigenspace of 3D shape vectors. Yet another alternative is to define an implicit probability model by sampling from the motions seen in the training data [22]. Currently, most human body trackers use particle filtering, and thus it is essential to be able to sample from the prior to generate new hypotheses. In the tree-based filter this is not necessary, and the dynamics can be encoded by transition probabilities between discrete states.

7 Results

We demonstrate the effectiveness of our technique by tracking both hand motion and finger articulation in cluttered scenes using a single camera. The results reveal the ability of the tree-structure to handle ambiguity arising from self-occlusion and 3D motion.

7.1 3D Hand Tracking Experiments

In two sequences we track the global 3D motion of the hand without finger articulation. The 3D rotations are limited to a hemisphere. At the leaf level, the tree has the following resolutions: 15 degrees in two 3D rotations, 10 degrees in image restoration and 5 different scales. These 12960 templates are then combined with a search at 2-pixel resolution in the image translation space. Figures 8 and 7 show results from tracking a pointing and an open hand, respectively, through their global motions.

In the third experiment the tree is built by hierarchical k-means clustering of a training data set of size 7200, The data set was captured while performing similar gestures as shown in figure 8 a number of times and in different order. In this case the tree has 7200 nodes

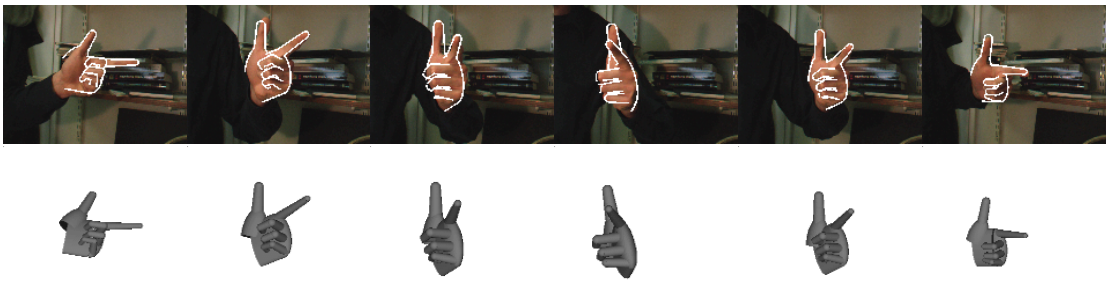


Figure 6: **Tracking a pointing hand in front of clutter.** *The images are shown with projected contours superimposed (top) and corresponding 3D avatar models (bottom), which are estimated using the tree-based algorithm. The hand is translating and rotating. A 2D deformable template would have problems coping with topological shape changes caused by self-occlusion.*

(i.e. all data points are used) at the leaf level, which is combined with a search at 2-pixel resolution in the image translation space. Transition probabilites between the nodes are learned from the training data. In another experiment the tree is built by partitioning a lower dimensional eigenspace. Applying PCA to the data set shows that more than 96 percent of the variance is captured within the first four principal components, thus we partition the four dimensional eigenspace. The number of nodes at the leaf level in this case is 9163. The results when using this method to build the tree are qualitatively very similar to clustering on the same test sequence and are not shown.

In the fourth sequence (figure 9) tracking is demonstrated for global hand motion together with finger articulation. The articulation parameters for the thumb and fingers are approximated by 7 and 5 divisions, 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.

Note that in all cases, the hand model was automatically initialised by searching the complete tree in the first frame of the sequence.

7.2 Comparison with Particle Filtering

We ran a standard particle filter on the first three sequences shown here, using up to 10000 particles. The same likelihood, prior and transition distributions are used. The filter tracks for about twenty frames, but then loses lock. One of the issues in particle filtering is the dependency on the generation of good hypotheses, which relies on having (a) a good spread of particles at time $(t-1)$ and (b) a good diffusion of particles to time t. Due to the stochastic nature of the algorithm, neither of these can be guaranteed unless the number of particles is exponentially large. The tree-based search allows an efficient deterministic search of the state space such that whole regions of low likelihood can be discarded with little computational effort.

A further disadvantage of using particle filtering is that the hypotheses have to be generated on-line. In our case projecting the 3D model contours with occlusion handling, is approximately 1000 times more expensive than evaluating the likelihood function. Currently we can generate 100 templates per second, which is the limiting factor for an on-line

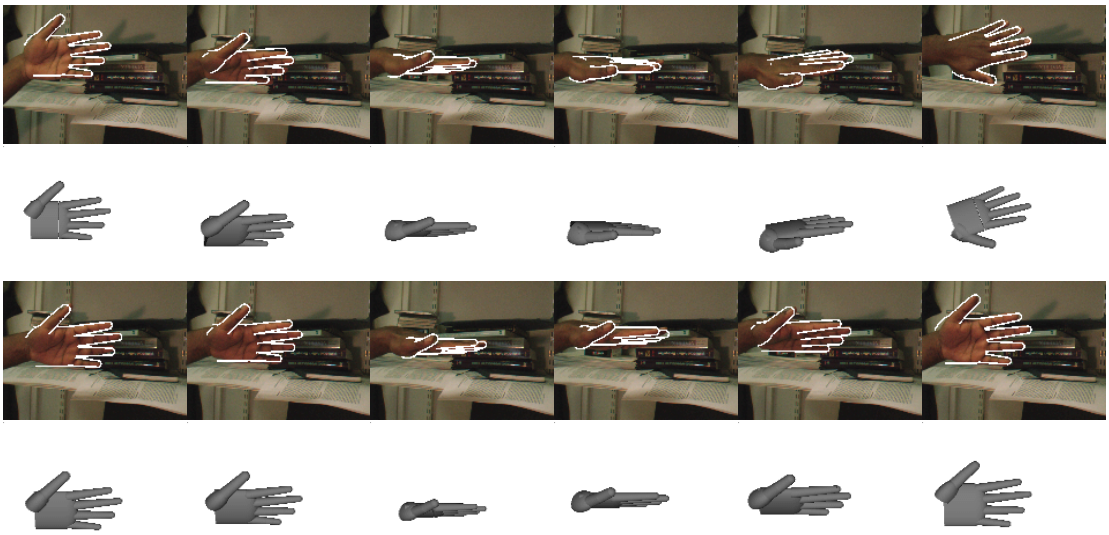


Figure 7: **Tracking a flat hand rotating in clutter.** *In this sequence the hand undergoes rotation and translation. The frames showing the hand with significant self-occlusion do not provide much data, and template matching becomes unreliable. By including prior information, these situations can be resolved. The projected contours are superimposed on the images, and the corresponding 3D avatar model, which is estimated using the tree-based algorithm, is shown below each frame.*

method. As the templates are generated only once for the tree-based method, this step can be done off-line. Running a particle filter with 10000 particles requires nearly two minutes per frame, whereas the tree-based filter, takes less on average 2 seconds per frame, typically involving an average of 9000 likelihood evaluations.

8 Summary and Conclusion

This paper endeavours 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 filtering using a tree-based estimator (FILTOR) 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.

Acknowledgements The authors would like to thank the Gottlieb Daimler–and Karl Benz–Foundation, the Engineering and Physical Sciences Research Council, the Gates Cambridge Trust, and the Overseas Research Scholarship Programme for their support.

References

[1] K. N. An, E. Y. Chao, W. P. Cooney, and R. L. Linscheid. Normative model of human hand for biomechanical analysis. *J. Biomechanics*, 12:775–788, 1979.

[2] V. Athitsos and S. Sclaroff. An appearance-based framework for 3D hand shape classification and camera viewpoint estimation. In *IEEE Conference on Face and Gesture Recognition*, pages 45–50, Washington D.C., USA, May 2002.

[3] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. Conf. Computer Vision and Pattern Recognition*, Madison, USA, June 2003. to appear.

[4] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.

[5] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications.* PhD thesis, Linköping University, Linköping, Sweden, 1999.

[6] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intell.*, 10(6):849–865, November 1988.

[7] R. S. Bucy and K. D. Senne. Digital synthesis of nonlinear filters. *Automatica*, (7):287–298, 1971.

[8] K. Choo and D. J. Fleet. People tracking using hybrid monte carlo filtering. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 321–328, Vancouver, Canada, July 2001.

[9] A. Doucet, N. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, 2001.

[10] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. 6th European Conf. on Computer Vision*, volume II, pages 37–49, Dublin, Ireland, June/July 2000.

[11] D. M. Gavrila and L. S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, June 1996.

[12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to non-linear and non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993.

[13] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, pages 93–101, Berlin, Germany, May 1993.

[14] M. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. on Computer Vision*, pages 343–356, Cambridge, UK, April 1996.

[15] I. A. Karaulova, P. M. Hall, and A. D. Marshall. A hierarchical model of dynamics for tracking people with a single video camera. In *Proc. British Machine Vision Conference*, volume 1, pages 352–361, Bristol, UK, September 2000.

[16] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision*, volume 2, pages 3–19, Dublin, Ireland, June 2000.

[17] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[18] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *Transactions on Image Processing*, 6(1):103–113, January 1997.

[19] V. Pavlović, J. M. Rehg, T.-J. Cham, and K. P. Murphy. A dynamic bayesian network approach to tracking using learned learned dynamic models. In *Proc. 7th Int. Conf. on Computer Vision*, pages 366–380, Corfu, Greece, September 1999.

[20] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *Proc. 3rd European Conf. on Computer Vision*, volume II, pages 35–46, May 1994.

[21] N. Shimada, K. Kimura, and Y. Shirai. Real-time 3-D hand posture estimation based on 2-D appearance retrieval using monocular camera. In *Proc. Int. WS. RATFG-RTS*, pages 23–30, Vancouver, Canada, July 2001.

[22] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. 7th European Conf. on Computer Vision*, volume 1, pages 784–800, Copenhagen, Denmark, May 2002.

[23] H. W. Sorenson. *Bayesian Analysis of Time Series and Dynamic Models*, chapter Recursive Estimation for nonlinear dynamic systems. Marcel Dekker inc., 1988.

[24] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 310–315, Kauai, USA, December 2001.

[25] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[26] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proc. Conf. Computer Vision and Pattern Recognition*, Madison, USA, June 2003. to appear.

[27] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. Journal of Computer Vision*, pages 9–19, June 2002.

[28] Y. Wu and T. S. Huang. Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. 7th Int. Conf. on Computer Vision*, volume I, pages 606–611, Corfu, Greece, September 1999.

[29] Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *Proc. 8th Int. Conf. on Computer Vision*, volume II, pages 426–432, Vancouver, Canada, July 2001.

[30] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. In *Proc. 3rd Asian Conf. on Computer Vision*, Hong Kong, China, January 1998.

Figure 8: **Tracking articulated hand motion in front of a cluttered background.** *In this sequence a number of different finger motions are tracked. The images are shown with projected contours superimposed (top) and corresponding 3D avatar models (bottom), which are estimated using the tree-based filter. The nodes in the tree are found by hierarchical clustering of training data in the parameter space, and dynamic information is encoded as transition probabilities between the clusters.*

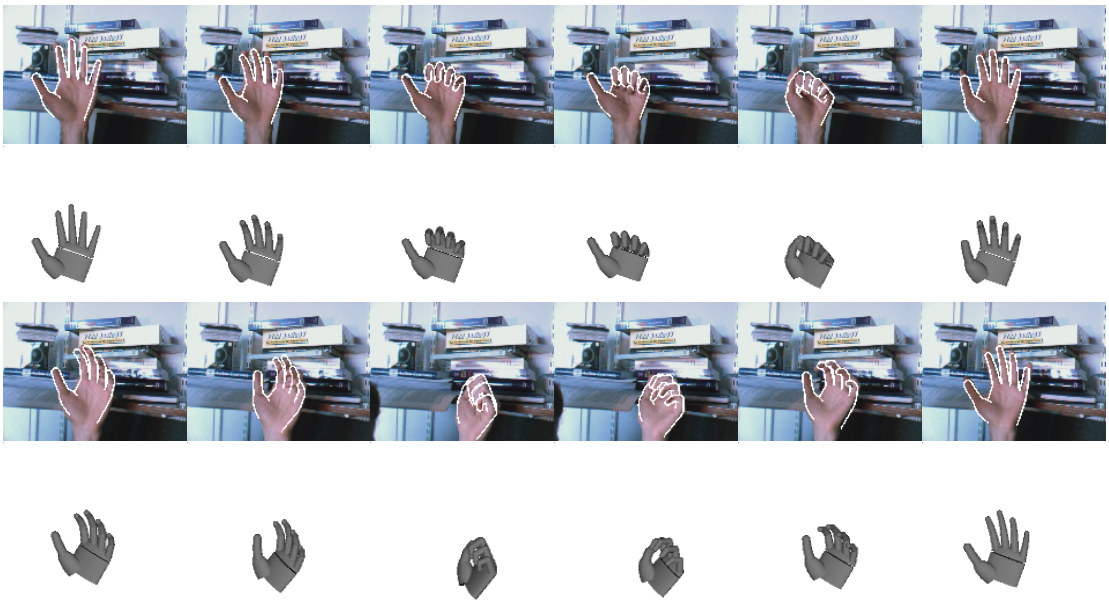


Figure 9: **Tracking a hand opening and closing with rigid body motion in front of a cluttered background.** *This sequence is challenging because the hand undergoes translation and rotation while opening and closing the fingers. 6* DOF *for rigid body motion plus 2* DOF *using manifolds for finger flexion and extension are tracked successfully with the tree-based algorithm.*