Model-Based Hand Tracking Using a Hierarchical Bayesian Filter

Björn Stenger [*], Arasanathan Thayananthan [†], Philip H. S. Torr [‡], and Roberto Cipolla [§]

September 19, 2006

Abstract

This paper sets out a tracking framework, which is applied to the recovery of three-dimensional hand motion from an image sequence. The method handles the issues of initialization, tracking, and recovery in a unified way. In a single input image with no prior information of the hand pose, the algorithm is equivalent to a hierarchical detection scheme, where unlikely pose candidates are rapidly discarded. In image sequences a dynamic model is used to guide the search and approximate the optimal filtering equations. A dynamic model is given by transition probabilities between regions in parameter space and is learned from training data obtained by capturing articulated motion. The algorithm is evaluated on a number of image sequences, which include hand motion with self-occlusion in front of a cluttered background.

Keywords: Probabilistic algorithms, video analysis, tracking.

1 Introduction

One of the fundamental problems in vision is that of tracking objects through sequences of images. Within this paper we present a Bayesian algorithm for tracking the 3D position and orientation of rigid or non-rigid objects. The application considered here is tracking hands in monocular video sequences, but the method is equally applicable to full body tracking [28, 29]. Great strides have been made in the theory and practice of tracking, for example the development of particle filters recognized that a key aspect in tracking was a better representation of the posterior distribution of model parameters [11, 17, 20, 36, 37]. Particle filters go beyond the uni-modal

[*]B. Stenger is with the Toshiba Corporate R&D Center, Kawasaki 212-8582, Japan. E-mail: bjorn@cantab.net.

[†]A. Thayananthan is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK. E-mail: at315@cam.ac.uk.

[‡]P.H.S. Torr is with the Department of Computing, Oxford Brookes University, Oxford OX33 1HX, UK. E-mail: philiptorr@brookes.ac.uk.

[§]R. Cipolla is with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK. E-mail: cipolla@eng.cam.ac.uk.

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 [10, 27]. In addition, even for low dimensional spaces there is a tendency for particles to become concentrated in a single mode of the distribution [12] and the tracker's stability mostly relies on the quality of the importance sampler.

Within this paper we consider tracking an articulated hand in cluttered images, without the use of markers. In general this motion has 27 degrees of freedom (DOF), 21 DOF for the joint angles and 6 for orientation and location [13, 32]. This state space can be reduced by reparameterization. Wu *et al.* [45] show that due to the correlation of joint angles, the state space for the joints can be approximated with 7 DOF by applying PCA, however the tracker has difficulties dealing with out-of-plane rotations and scale changes.

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 [16] 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 [27] 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 [1, 2, 33, 34]. 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 pedestrians [15]. 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, 19], 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 [15, 31]. 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 for complete image frames by Jojic *et al.* [24] and for exemplar templates by Toyama and Blake [43]. 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.*" This paper presents one solution, which combines ideas from hierarchical view-based detection and probabilistic tracking in the object parameter space. A large number of templates are generated from a 3D model and a hierarchy of these templates is constructed off-line by partitioning the parameter space. The finest partition corresponds to the

leaves of the tree. At each time instant the posterior distribution of the state parameters is estimated over these partitions. If no dynamic information is available, for example in the first frame of a sequence, this corresponds to a hierarchical detection scheme. In subsequent frames, the distribution is propagated over time while making use of global and intrinsic object dynamics.

2 Hierarchical Filtering

This section proposes an algorithm for Bayesian tracking, which is based on a multi-resolution partitioning of the state space. It is motivated by methods introduced in the context of hierarchical object detection, which are briefly outlined in the next section.

2.1 Tree-based detection

Methods for detecting objects are becoming increasingly efficient. Examples are real-time face detection or pedestrian detection [15, 44], both of which are based on hierarchical or cascaded methods. However, applying these techniques to hand detection from a single image is difficult because of the large variation in shape and appearance of a hand in different poses. In this case detection and pose estimation are tightly coupled. One approach to solving this problem is to use a large number of shape templates and find the best match in the image. In *exemplar-based* methods, such templates are obtained directly from the training sets [8, 15, 26, 43]. For example, Gavrila uses approximately 4500 shape templates to detect pedestrians in images [15]. To avoid exhaustive search, a template hierarchy is formed by bottom-up clustering based on the chamfer distance. A number of similar shape templates are represented by a cluster prototype. This prototype is first compared to the input image, and only if the error is below a threshold value, are the templates within the cluster compared to the image. The use of a template hierarchy is reported to result in a speed-up of three orders of magnitude compared to exhaustive matching [15].

If a parametric object model is available, another option to build such a tree of templates 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 $\mathcal{P}_l = \{\mathcal{S}^{i,l}\}_{i=1}^{N_l}$. The leaves of the tree define the finest partition of the state space $\mathcal{P}_L = \{\mathcal{S}^{i,L}\}_{i=1}^{N_L}$. The use of a parametric model also allows the combination of a a template hierarchy created off-line with an on-line optimization process. Once the leaf level is reached, the model can be refined by continuous optimization of the model's parameters. In [42] we used this method to adapt the shape of a generic hand model to an individual user in the first frame.

A draw-back of a single-frame exemplar based detector, such as the one presented in [15], is the difficulty of incorporating temporal constraints. We take inspiration from Jojic *et al.* [24] who modeled a video sequence by a small number of image exemplars and modeled the motions by a discrete label set, imposing dynamics by a hidden Markov model. This idea was taken further by Toyama and Blake [43] who suggested a metric mixture model for exemplar-based tracking. The integration of a dynamic model is useful, firstly to resolve ambiguous situations, and secondly, to smooth the motion. However, in [43] no template hierarchy is formed as the problem is not seen as one of efficient object detection. The following section introduces an algorithm which combines the efficiency of hierarchical methods with Bayesian filtering.

2.2 Tree-based filtering

Tracking is formulated as a Bayesian inference problem, where the internal parameters of an object at time t are given by values $\mathbf{x}_t \in \mathbb{R}^n$ of a random variable $\mathcal{X}_t$, and the measurement obtained are values $\mathbf{z}_t \in \mathbb{R}^m$ of the random variable $\mathcal{Z}_t$. Given the observations up to and including time t, $\mathbf{z}_{1:t} = \{\mathbf{z}_i\}_{i=1}^t$, the state estimation is expressed as a pair of recursive prediction and update equations [23]:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})\, d\mathbf{x}_{t-1} \tag{1}$$

$$\text{and} \quad p(\mathbf{x}_t|\mathbf{z}_{1:t}) = c_t^{-1}\, p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1})\,, \tag{2}$$

$$\text{where} \quad c_t = \int p(\mathbf{z}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{z}_{1:t-1})\, d\mathbf{x}_t. \tag{3}$$

In the general case it is not possible to obtain analytic solutions for these equations, but there exist a number of approximation methods which can be used to obtain a numerical solution [12, 38].

An important issue in each approach is how to represent the prior and posterior distributions in the filtering equations. One suggestion, introduced by Bucy and Senne [9], is to use a point-mass representation on a uniform grid. The grid is defined by a discrete set of points in state space, and is used to approximate the integrals in the filtering equations by replacing continuous integrals with Riemann sums over finite regions. The distributions are approximated as piecewise constant over these regions. The underlying assumption of this method is that the posterior distribution is band-limited, so that there are no singularities or large oscillations between the points on the grid. Typically grid-based filters have been applied using an evenly spaced grid and the evaluation is thus exponentially expensive as the dimension of the state space increases [5, 9]. Bucy and Senne suggest modeling each mode of the distribution by a separate adapting grid, and they devise a scheme for creating and deleting local grids on-line. A different approach is taken by Bergman [5], who uses a fixed grid, but avoids the evaluation at grid points where the probability mass is below a threshold value.

The aim in this section is to design an algorithm that can take advantage of the efficiency of tree-based search to efficiently compute an approximation to the optimal Bayesian solution using a grid-based filter. In the following, assume that the values of the state vectors $\mathbf{x} \in \mathbb{R}^n$ are within a compact region $\mathcal{R}$ of the state space. In the case of hand tracking, this corresponds to the fact that the parameter values are bounded, the boundary values being defined by the valid range of motion. Define a multi-resolution partition of the region $\mathcal{R}$ as described in section II-A by dividing the region $\mathcal{R}$ at each tree-level l into N_l partitions $\{\mathcal{S}^{i,l}\}_{i=1}^{N_l}$,

$$\bigcup_{i=1}^{N_l} \mathcal{S}^{i,l} = \mathcal{R} \quad \text{for } l = 1, \ldots, L\,. \tag{4}$$

A graphical depiction is shown in figure 1. The posterior distribution is represented as piecewise constant over these sets, the distribution at the leaf level being the representation at the highest resolution.

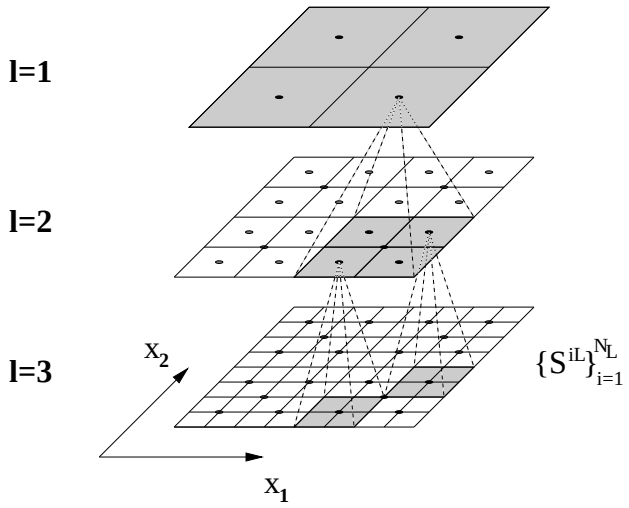


Figure 1: **Hierarchical partitioning of the state space.** *The state space is partitioned using a multi-resolution grid. The regions $\{\mathcal{S}^{i,L}\}_{i=1}^{N_L}$ at the leaf level define the finest partition, over which the filtering distributions are approximated as piecewise constant. The number of regions is exponential in the state dimension. However, if large regions of parameter space have negligible probability mass, these can be identified early, achieving reduction in computational complexity.*

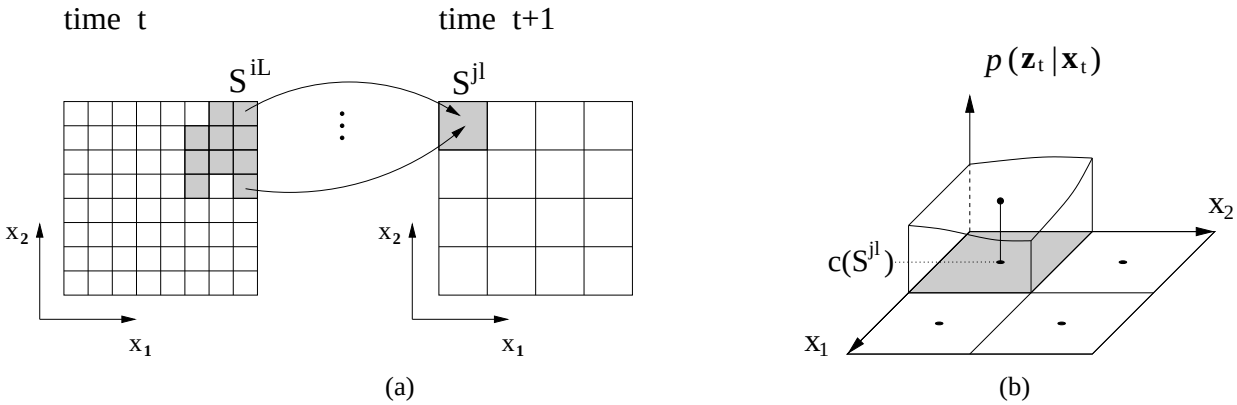


Figure 2: **Discretizing the filtering equations. (a)** *The transition distributions are approximated by transition probabilities between discrete regions in state space, which can be modeled by a Markov transition matrix.* **(b)** *The likelihood function is evaluated at the centre $c(\mathcal{S}^{j,l})$ of each region, assuming that the function is locally smooth.*

Define a discrete probability distribution $p(\hat{\mathbf{x}}_t^{i,l})$ over the regions $\mathcal{S}^{i,l}$,

$$p(\hat{\mathbf{x}}_t^{i,l}|\mathbf{z}_{1:t}) = \int_{\mathbf{x}_t \in \mathcal{S}^{i,l}} p(\mathbf{x}_t|\mathbf{z}_{1:t})\, d\mathbf{x}_t. \tag{5}$$

In the first frame, the posterior is set to the likelihood distribution. In the following frames the discrete recursive relations are obtained from the continuous case by integrating over regions. Given the distribution over the leaves of the tree, $p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1})$, at the previous time step $t-1$,

the prediction equation now becomes a transition between discrete regions $\mathcal{S}^{i,L}$ and $\mathcal{S}^{j,l}$ in state space:

$$p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t-1}) = \sum_{i=1}^{N_L} p(\hat{\mathbf{x}}_t^{j,l}|\hat{\mathbf{x}}_{t-1}^{i,L})\ p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1}). \tag{6}$$

Given the state transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ the transition probabilities are approximated by region to region transition probabilities, see figure 2a. In order to evaluate the distribution $p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t})$, the likelihood $p(\mathbf{z}_t|\hat{\mathbf{x}}_t^{j,l})$ needs to be evaluated for a region in parameter space. This is computed by evaluating the likelihood function at a single point, taken to be the centre of the region $c(\mathcal{S}^{j,l})$. This approximation assumes that the likelihood function in the region $\mathcal{S}^{j,l}$ can be represented by the value at the centre location $c(\mathcal{S}^{j,l})$, see figure 2b.

$$p(\mathbf{z}_t|\hat{\mathbf{x}}_t^{j,l}) \ \propto\ (\mathbf{z}_t|c(\mathcal{S}^{j,l}))\,. \tag{7}$$

This is similar to the idea of using a cluster prototype to represent similar shapes.

Having laid out Bayesian filtering over discrete states, the question arises how to combine the theory with the efficient tree-based algorithm previously described. The idea is to approximate the posterior distribution by evaluating the filter equations at each level of the tree. In a breadth-first traversal regions with low probability mass are identified and not further investigated at the next level of the tree. Regions with high posterior are explored further in the next level (Figure 3). It is expected that the higher levels of the tree will not yield accurate approximations to the posterior, but are used to discard inadequate hypotheses, for which the posterior of the set is below a threshold value. In the experiments the template hierarchy is built by manually setting the resolution for each parameter dimension such that the appearance within each region is below a threshold value. At each level of the tree the maximum $\hat{p}_t^{l,max}$ and minimum $\hat{p}_t^{l,min}$ of the posterior values is computed and the threshold is chosen as

$$\tau_t^l = \hat{p}_t^{l,min} + c_\tau\ (\hat{p}_t^{l,max} - \hat{p}_t^{l,min})\ . \tag{8}$$

where $c_\tau = 0.5$ in our experiments. Alternatively, to fix the computation time, only a constant number of modes could be explored. By changing the threshold value, the trade-off between accuracy and computational time can be regulated. Note that the local maxima on one level do not necessarily correspond to the global maxima of the posterior distribution. In particular, if τ is set too high, the branch containing the global maximum may be missed, leading to an incorrect pose estimate in that frame. After each time step t the posterior distribution is represented by the piecewise constant distribution over the regions at the leaf level. When a hand is in the scene, this leads to a distribution where there is at least one strong peak, whereas in background scenes the values do not vary as much. In this case no dynamic information is used and as in the initialization step only the likelihood values are computed at the first level. An overview of the algorithm is given in Algorithm 1.

2.3 Edge and color likelihoods

This section introduces the likelihood function which is used within the algorithm. The likelihood $p(\mathbf{z}|\mathbf{x})$ relates observations $\mathbf{z}$ in the image to the unknown state $\mathbf{x}$. The observations are based on

Algorithm 1 Tree-based filtering equations

Notation: $par(j)$ denotes the parent of node j.

Initialization step at $t = 1$, assuming uniform distribution over the states initially.

At level $l = 1$:
$$p(\hat{\mathbf{x}}_1^{j,1}|\mathbf{z}_1) = p(\mathbf{z}_1|\hat{\mathbf{x}}_1^{j,1}) \quad \text{for } j = 1, \ldots, N_1\,. \tag{9}$$

At level $l > 1$:
$$p(\hat{\mathbf{x}}_1^{j,l}|\mathbf{z}_1) = \begin{cases} p(\mathbf{z}_1|\hat{\mathbf{x}}_1^{j,l}) & \text{if } \; p(\hat{\mathbf{x}}_1^{par(j),l-1}|\mathbf{z}_1) > \tau_t^{l-1}, \\ p(\hat{\mathbf{x}}_1^{par(j),l-1}|\mathbf{z}_1) & \text{otherwise,} \end{cases} \tag{10}$$

Normalize after computing the values at each level l such that $\sum_{j=1}^{N_l} p(\hat{\mathbf{x}}_1^{j,l}|\mathbf{z}_1) = 1$.

At time $t > 1$

At level $l = 1$:
$$p(\hat{\mathbf{x}}_t^{j,1}|\mathbf{z}_{1:t}) = p(\mathbf{z}_t|\hat{\mathbf{x}}_t^{j,1})\, p(\hat{\mathbf{x}}_t^{j,1}|\mathbf{z}_{1:t-1})\,, \tag{11}$$

where
$$p(\hat{\mathbf{x}}_t^{j,1}|\mathbf{z}_{1:t-1}) = \sum_{i=1}^{N_L} p(\hat{\mathbf{x}}_t^{j,1}|\hat{\mathbf{x}}_{t-1}^{i,L})\; p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1}) \tag{12}$$

At level $l > 1$:
$$p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t}) = \begin{cases} p(\mathbf{z}_t|\hat{\mathbf{x}}_t^{j,l})\, p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t-1}) & \text{if } \; p(\hat{\mathbf{x}}_t^{par(j),l-1}|\mathbf{z}_{1:t}) > \tau_t^{l-1}, \\ p(\hat{\mathbf{x}}_t^{par(j),l-1}|\mathbf{z}_{1:t}) & \text{otherwise,} \end{cases} \tag{13}$$

where
$$p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t-1}) = \sum_{i=1}^{N_L} p(\hat{\mathbf{x}}_t^{j,l}|\hat{\mathbf{x}}_{t-1}^{i,L})\; p(\hat{\mathbf{x}}_{t-1}^{i,L}|\mathbf{z}_{1:t-1})\,. \tag{14}$$

Normalize after computing the values at each level l such that $\sum_{j=1}^{N_l} p(\hat{\mathbf{x}}_t^{j,l}|\mathbf{z}_{1:t}) = 1$.

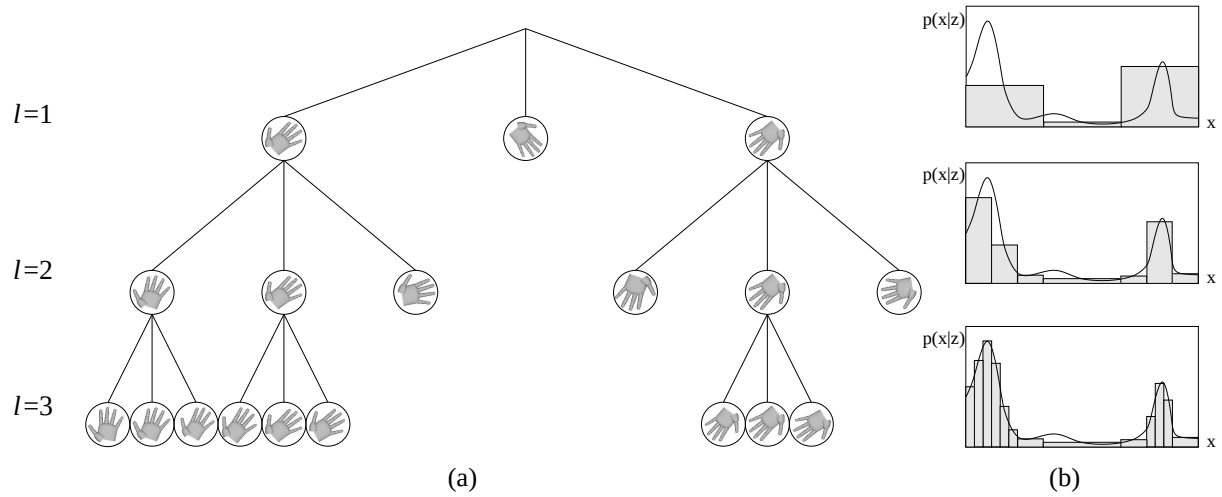


Figure 3: **Tree-based estimation of the posterior density. (a)** *Associated with the nodes at each level is a non-overlapping set in the state space, defining a partition of the state space. The posterior for each node is evaluated using the center of each set, depicted by a hand rotated by a specific angle. Sub-trees of nodes with low posterior are not further evaluated.* **(b)** *Corresponding posterior density (continuous) and the piecewise constant approximation. The modes of the distribution are approximated with higher precision at each level.*

the edge map $\mathbf{z}^{edge}$ of the image, as well as pixel color values $\mathbf{z}^{col}$. These features have proved useful for detecting and tracking hands in previous work [26, 27, 46]. In the following sections the joint likelihood of $\mathbf{z} = (\mathbf{z}^{edge}, \mathbf{z}^{col})^{\mathrm{T}}$ is approximated as

$$p(\mathbf{z}|\mathbf{x}) \;=\; p(\mathbf{z}^{edge}, \mathbf{z}^{col} \,|\, \mathbf{x}) \;\approx\; p(\mathbf{z}^{edge} \,|\, \mathbf{x})\; p(\mathbf{z}^{col} \,|\, \mathbf{x})\,, \tag{15}$$

thus treating the observations independently. The likelihood term for each of the observations is derived in the following sections.

Edge likelihood: The edge likelihood term $p(\mathbf{z}^{edge}|\,\mathbf{x})$ is based on the chamfer distance function [4, 7]. Given the set of template points $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^{N_a}$ and the set of Canny edge points $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^{N_b}$, a quadratic chamfer distance function is given by the average of the squared distances between each point of $\mathcal{A}$ and its closest point in $\mathcal{B}$:

$$d(\mathcal{A}, \mathcal{B}) \;=\; \frac{1}{N_a} \sum_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} ||\, a - b \,||^2. \tag{16}$$

The chamfer function can be computed efficiently for many model templates by using a distance transform (DT) of the edge image. This transformation takes the set of feature points $\mathcal{B}$ as input and assigns each location the distance to its nearest feature, i.e. the DT value at location u contains the value $\min_{b \in \mathcal{B}} ||\, u - b \,||$. The chamfer function for a single template can be computed by correlating its points with the DT image. To increase robustness toward partial occlusion the DT image is thresholded by an upper bound τ on the distance to the edge, typically $\tau = 20$. Edge

orientation is included by decomposing both template and edge image into a number of separate orientation channels according to gradient orientation. The distance is computed separately for each channel, thereby increasing the discriminatory power of the likelihood function, especially in cases when there are many background edge points present in the image [31]. The used cost term is thus

$$d_e(\mathcal{A}, \mathcal{B}) \;=\; \frac{1}{N_a} \sum_{i=1}^{N_\gamma} \sum_{a \in \mathcal{A}^i} \min \left(\min_{b \in \mathcal{B}^i} ||a - b||^2, \tau \right) \tag{17}$$

where $\mathcal{A}^i$ and $\mathcal{B}^i$ are the feature points in orientation channel i, and $N_\gamma = 6$, in our experiments. A shape template is treated as the centre of a mixture distribution, each component being a *metric exponential* distribution [43]. Given the shape template $\mathcal{P}$ and the observed edge image $\mathbf{z}^{edge}$, the likelihood function is defined as

$$p(\mathbf{z}^{edge} \mid \mathbf{x}) \;=\; \frac{1}{Z} \, \exp\left(-\lambda \, d_e(\mathcal{A}(\mathbf{x}), \mathcal{B}(\mathbf{z}^{edge}))\right) \,, \tag{18}$$

where $\mathcal{A}(\mathbf{x})$ denotes that the set of template points $\mathcal{A}$ is generated by projecting the model using the state vector $\mathbf{x}$, and $\mathcal{B}$ is the set of edge points obtained from the edge image $\mathbf{z}^{edge}$. Another option is to define the likelihood function based on the *PDF projection theorem* [41], which incorporates information about the distribution of background edges as well and is perhaps better justified by theory.

Color likelihood: The color likelihood function $p(\mathbf{z}^{col}|\mathbf{x})$ is based on a skin color distribution p^s and a background color distribution p^{bg}, respectively. Given a state vector $\mathbf{x}$, corresponding to a particular hand pose, the pixels in the image are partitioned into a set of locations within the hand silhouette $\{k : k \in S(\mathbf{x})\}$ and outside this region $\{k : k \in \bar{S}(\mathbf{x})\}$. If pixel-wise independence is assumed, the likelihood function for the whole image can be factored as

$$p(\mathbf{z}^{col}|\mathbf{x}) \;=\; \prod_{k \in S(\mathbf{X})} p^s\left(I(k)\right) \prod_{k \in \bar{S}(\mathbf{X})} p^{bg}\left(I(k)\right) \tag{19}$$

where $I(k)$ is the color vector at location k in the image. When taking the logarithm, this term is converted into a sum. The evaluation can now be performed efficiently by computing a sum table (or integral image), $\mathbf{B}^{sum}$, which has the same size as the image and contains the cumulative sums along the x-direction:

$$\mathbf{B}^{sum}(x, y) = \sum_{i=1}^{x} \left(\log p^s(I(i, y)) - \log p^{bg}(I(i, y))\right) \,, \tag{20}$$

where in this equation the image I is indexed by its x and y-coordinates. This array only needs to be computed once and is then used to compute sums over areas by adding and subtracting values at points on the silhouette contour. Thus the computation time is proportional to the contour length. In the experiments skin color is modeled with a Gaussian distribution in (r, g)-space, for background pixels a uniform distribution is assumed. Note that if a distribution of background appearance can be obtained, this should be used, e.g. if there is a static scene.

For an illustrative example using single-frame detection, which shows how combining edge and skin color information facilitates detection when one of the features becomes unreliable, see figure 4. Color is very useful when edge information is unreliable due to many background edges, low intensity contrast or fast hand motion. On the other hand, edge information allows accurate matching when the hand is in front of skin colored background.

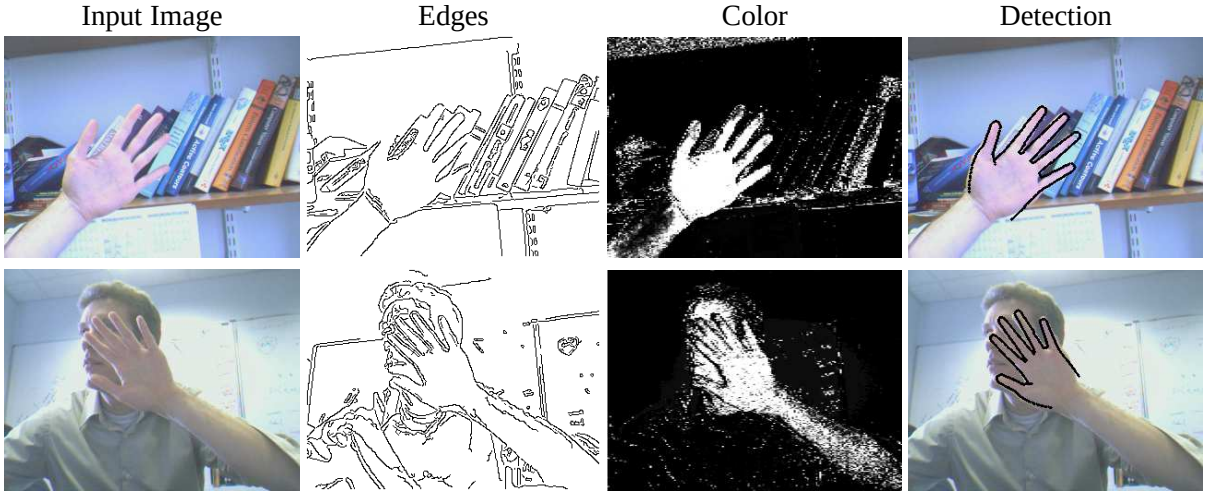


Figure 4: **Detection with integrated edge and color features. (Top row)** *Hand in front of cluttered background,* **(bottom row)** *hand in front of face; situations in which one of the cues is not discriminative (edges in row 1, color in row 2), but by using them in combination the hand is correctly detected in both cases (last column).*

In a second experiment on a sequence of 640 frames the hand pose was kept fixed as an open hand parallel to the image plane with 4 DOF motion; translation in x, y, and z-direction, as well as rotation around the z-axis. The task is made challenging by introducing a cluttered background with skin-colored objects. The hand motion is fast, and during the sequence the hand is partially and fully occluded, as well as out of the camera view. A set of 500 templates is generated, corresponding to 100 discrete orientations and five different scales, to search for the best match over the image. The translation space is sampled at a 6-pixel resolution. No dynamics are used in this sequence as the hand leaves and re-enters the camera view several times. Figure 5 shows typical results for a number of frames as well as the 2D position error measured against manually labelled ground truth. The RMS error over the complete sequence for the frames in which the hand was detected, was 3.7 pixels. For comparison, a single hypothesis version of the Kalman filter [39] was run on this sequence using a four dimensional state space and a first order dynamic model. The *unscented Kalman filter* (UKF) is a nonlinear extension of the Kalman filter [25]. The UKF uses an approximation of the underlying distributions using a set of sample points which are propagated through the original Kalman filter equations. The observation model used are local skin color edges, as in [27], i.e. points of transition between areas of high and low skin color likelihood. The UKF tracker was initialized manually in the first frame and tracked the hand for only 20 frames before lock was lost. The main reasons for this are that the color edge features alone are not robust enough and that the dynamic model is not able to handle fast and

abrupt motion. Different models (constant velocity and constant acceleration models) were also tested, but once the target was lost, the tracker was unable to recover.

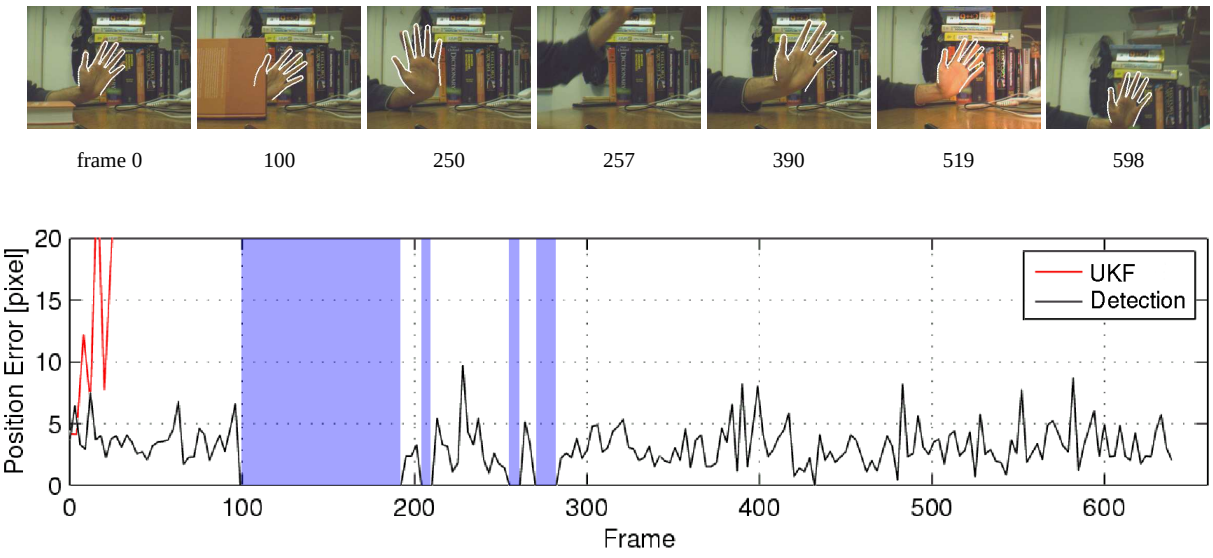


Figure 5: **Detecting an open hand.** *This figure shows successful detection using edge and color features of an open hand showing the best match superimposed, if the likelihood function is above a constant threshold. The sequence is challenging because the background contains skin-colored objects and motion is fast, leading to motion blur and missed edges. The method handles partial occlusion and lighting changes to some degree, can initialize and deal with unsteady camera motion. The graph shows an error plot for the detection algorithm and a Kalman filter (UKF) tracker. The hand position error was measured against manually labelled ground truth. The shaded areas indicate intervals in which the hand is either fully occluded or out of camera view. The detection algorithm successfully finds the hand in the whole sequence, whereas the UKF tracker using skin-color edges is only able to track the hand for a few frames. The reasons for the loss of track is that the hand motion is fast between two frames and that skin-color edges cannot be reliably found in this input sequence.*

2.4 Modeling hand dynamics

The global motion of the hand is modeled using a zero order Gaussian model, making only weak prior assumptions about motion continuity. Other models, such as a second order model learned from data [6] or a mixed-state tracker [22] have also been used for modeling global hand motion. However, as shown in the previous experiment, choosing a particular motion model can be restrictive.

Articulated motion is naturally constrained, since each joint can only move within certain limits and the motion of different joints is correlated [45]. Thus the articulation parameters are expected to lie within a compact region in the 21 dimensional angle space. The dynamics for this articulated motion are modeled as a first order process, which are learned from training data obtained from three subjects with a data glove. Since discrete regions in state space are

considered, the process can be described by a Markov transition matrix $\mathbf{M}^{LL} \in [0,1]^{N_L \times N_L}$, which contains the transition probabilities between the regions $\{S^{j,L}\}_{j=1}^{N_L}$ at the leaf-level. In order to evaluate the transitions at different tree levels, a transition matrix $\mathbf{M}^{Ll} \in [0,1]^{N_L \times N_l}$ for each level l of the tree is required, where each matrix contains the values:

$$\mathbf{M}_{i,j}^{Ll} = p(\hat{\mathbf{x}}_t^{j,l} | \hat{\mathbf{x}}_{t-1}^{i,L}),\ i = 1, \ldots, N_L,\ j = 1, \ldots, N_l. \tag{21}$$

In practice, these matrices are sparse and the non-zero values are stored in a look-up table.

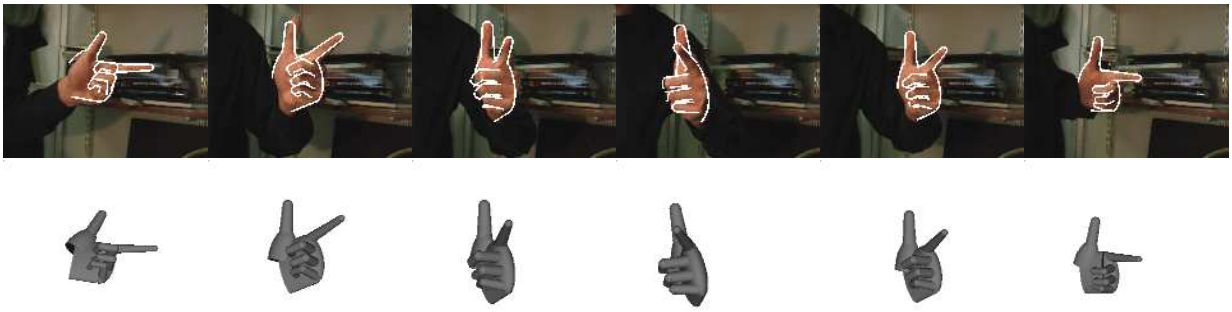


Figure 6: **Tracking a pointing hand**. *The images are shown with projected contours superimposed* **(top row)** *and corresponding 3D model* **(bottom row)**, *which are estimated using the tree-based filter. The hand is translating and rotating.*

3 Experimental Results

The following experiments show the effectiveness of the technique by detecting and tracking a hand in scenes using input from a single camera (image size 320×240).

3.1 Tracking rigid body motion

In the following experiments the hierarchical filter is used to track rigid hand motion in cluttered scenes using a single camera.

The results show the ability to tolerate self-occlusion during out-of-image-plane rotations. The 3D rotations are limited to a hemisphere and for each sequence 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 $13 \times 13 \times 19 \times 5 = 16,055$ templates. The resolution of the translation parameters is 20 pixels at the first level, 5 pixels on the second level, and 2 pixels at the leaf level. The translation is done by shifting the templates in the 2D image. Note that for each of the sequence a different tree is constructed, using the corresponding hand configuration. Figure 6 shows the tracking results on an input sequence of a pointing hand during translation and rotation. The top row shows the input frames with the projected model contours superimposed. The bottom row shows the corresponding pose of the 3D model. During the rotation there is self-occlusion as the index finger moves in front of the palm. For each frame the maximum a-posteriori (MAP) solution is shown. Figure 7 shows frames from the sequence of an open hand performing out of image plane rotation. This sequence contains 160 frames and

shows a hand first rotating approximately 180 degrees and returning to its initial pose (figure 7, two top rows). This is followed by a 90 degree rotation (figure 7, two bottom rows), and returning to its initial position. This motion is difficult to track as there is little information available when the palm surface is normal to the image plane.

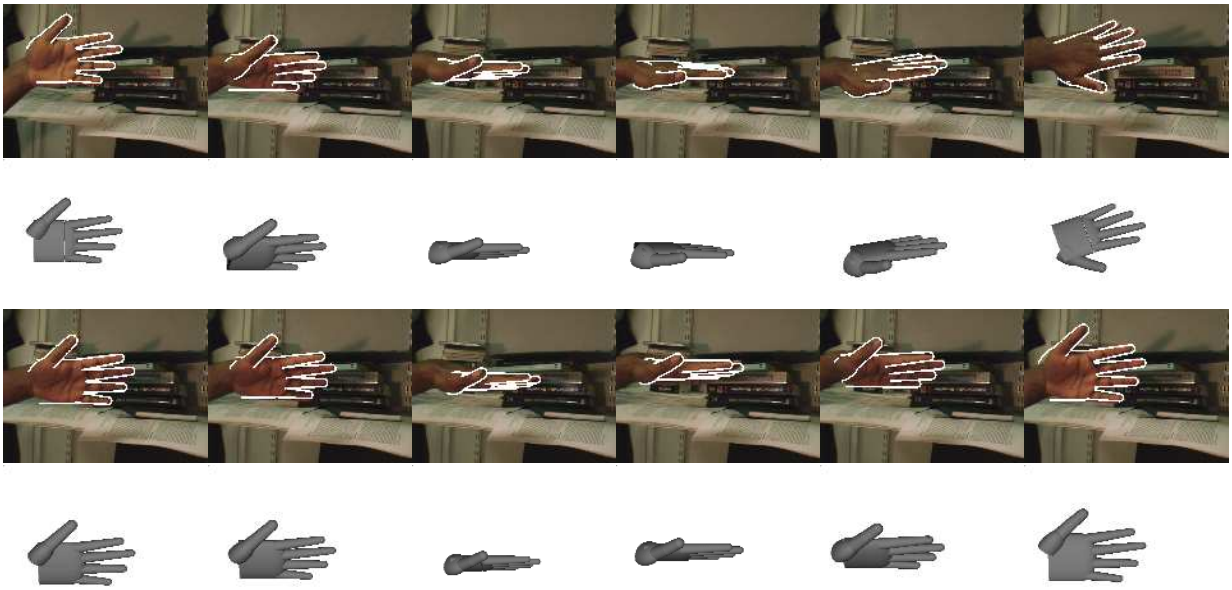


Figure 7: **Tracking out-of-image-plane rotation.** *In this sequence the hand undergoes rotation and translation. The frames showing the hand with self-occlusion do not provide much data, and template matching becomes unreliable. By including prior information, these situations can be resolved. The projected contours are superimposed on the images, and the corresponding 3D model is shown below each frame.*

Figure 8 shows example frames from a sequence of 509 frames of a pointing hand. The tracker handles fast motion and is able to recover after the hand has left the camera view and re-enters the scene. The computation takes approximately two seconds per frame on a 1GHz Pentium IV, corresponding to a speed-up of two orders of magnitude over exhaustive detection. Note that in all cases the hand model was automatically initialized in the first frame of the sequence.

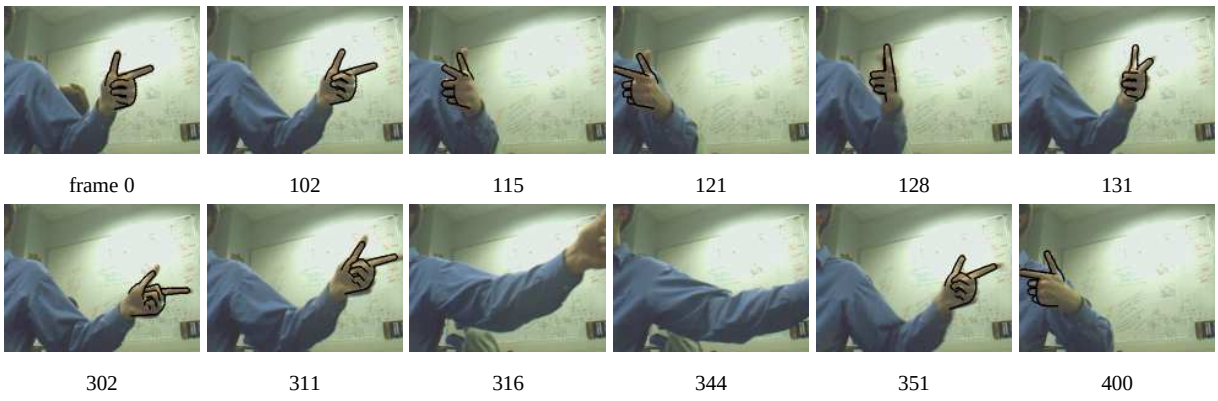


Figure 8: **Fast motion and recovery.** *This figure shows frames from a sequence tracking 6 DOF.* **(Top row)** *The hand is tracked during fast motion and* **(bottom row)** *the tracker is able to successfully recover after the hand has left the camera view.*

Figure 9 shows error plots for the three sequences. The error is the localization error measured against manually labelled ground truth locations of the tip of the thumb and one finger. It can be observed that the presence of peaks, which are due to local minima in the likelihood function, do not cause tracking to fail. The mean RMS error for the three sequences above is 6.7, 6.6, and 7.9 pixels, respectively.

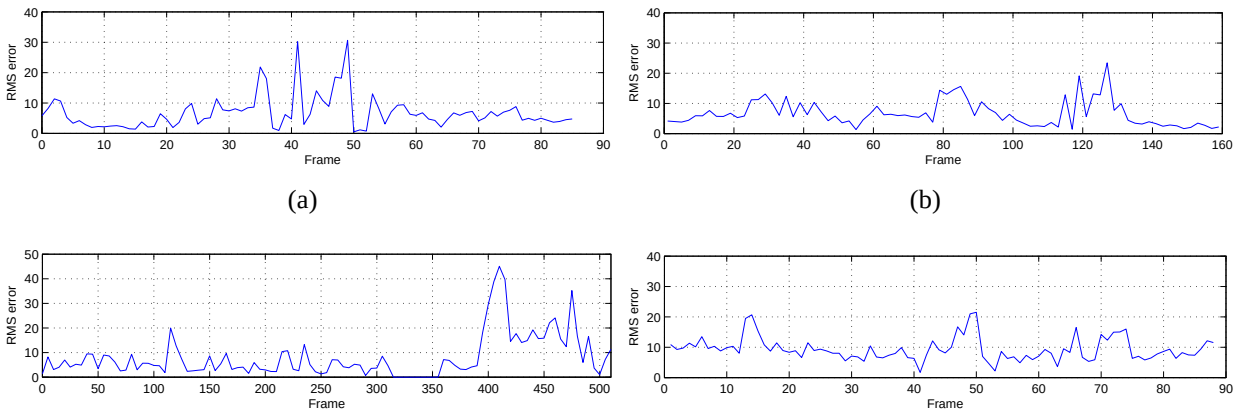


Figure 9: **Error performance.** *This figure shows the error performance in terms of finger tip localization measured against manually labelled ground truth.* **(a)** *pointing hand sequence of figure 6,* **(b)** *rotating hand sequence of figure 7,* **(c)** *pointing hand sequence of figure 8, where the hand is out of the view in frames 319–352,* **(d)** *opening and closing hand sequence of figure 15.*

3.2 Initialization

Two illustrative experiments demonstrate the algorithm during the initialization phase. In the first frame of each sequence the posterior term $p(\hat{\mathbf{x}}_1^{j,1}|\mathbf{z}_1)$ for each region is proportional to the likelihood value $p(\mathbf{z}_1|\hat{\mathbf{x}}_1^{j,1})$ for the first observation $\mathbf{z}_1$. A tree with four levels and 8748 templates

of a pointing hand at the leaf level was generated, corresponding to a search over 972 discrete angles and nine scales, and a search over translation space at single pixel resolution. As before, the motion is restricted to a hemisphere. The search process at different levels of the tree is illustrated in figure 10. The templates at the higher levels correspond to larger regions of parameter space, and are thus less discriminative. The image regions of the face and the second hand, for which there is no template, have relatively low cost as they contain skin color pixels and edges. As the search proceeds, these regions are progressively eliminated, resulting in only few final matches. Figure 11 gives a more detailed view by showing examples of templates at different tree levels which are above (accepted) and below (rejected) the threshold values $\tau_0^l, l = 1, \ldots, L,$ in equation (8). It can be seen that the templates which are above the threshold at the first level do not present very accurate matches, however a large number of templates can be rejected at this stage. At lower tree levels the accuracy of the match increases.

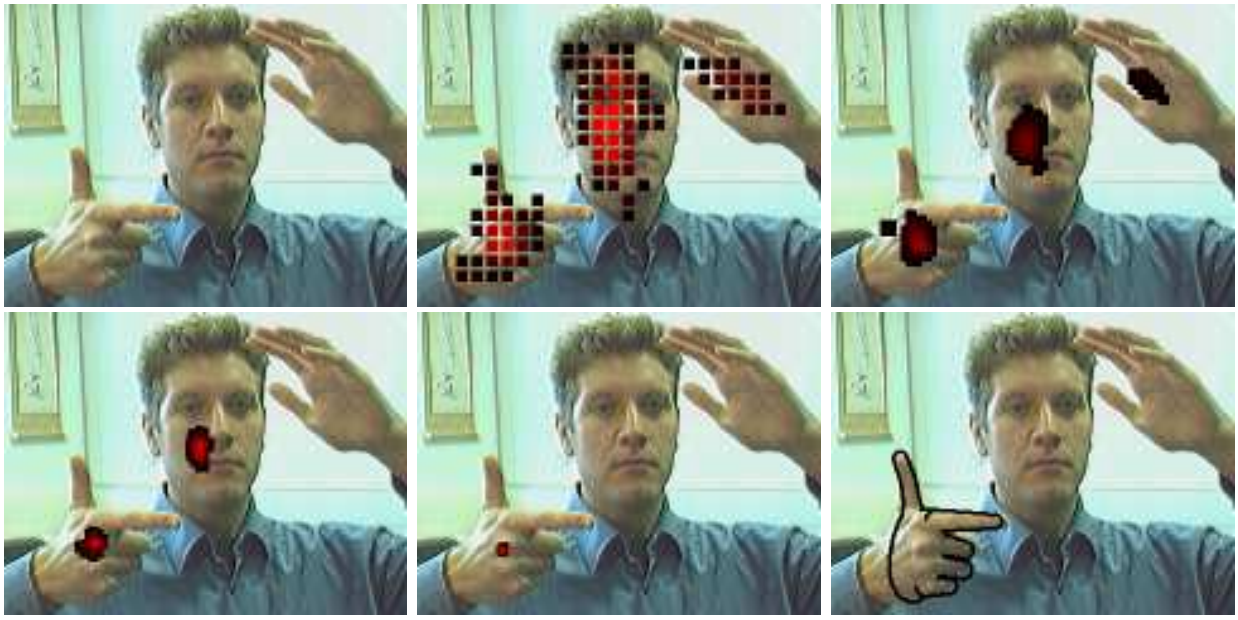


Figure 10: **Automatic initialization.** *From* **Top Left***: Input image,* **next***: Images with detection results super-imposed. Each square represents an image location which contains at least one node with a likelihood estimate above the threshold value. The intensity indicates the number of matches, high intensity indicated larger number of matches. Ambiguity is introduced by the face and the second hand. Regions are progressively eliminated, the best match is shown on the bottom right.*

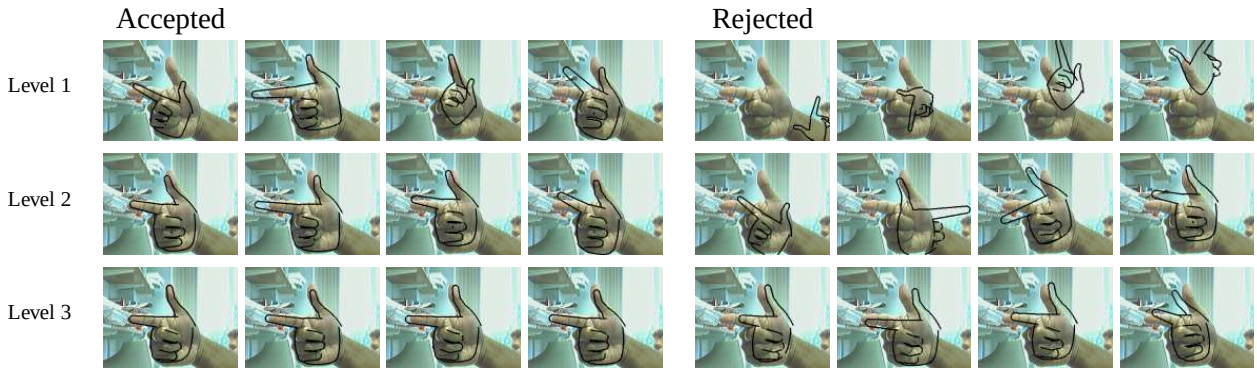


Figure 11: **Search results at different levels of the tree.** *This figure shows templates above and below the threshold values τ_t^l at levels 1 to 3 of the tree, ranked according to their likelihood values. As the search is refined at each level, the difference between accepted and rejected templates decreases.*

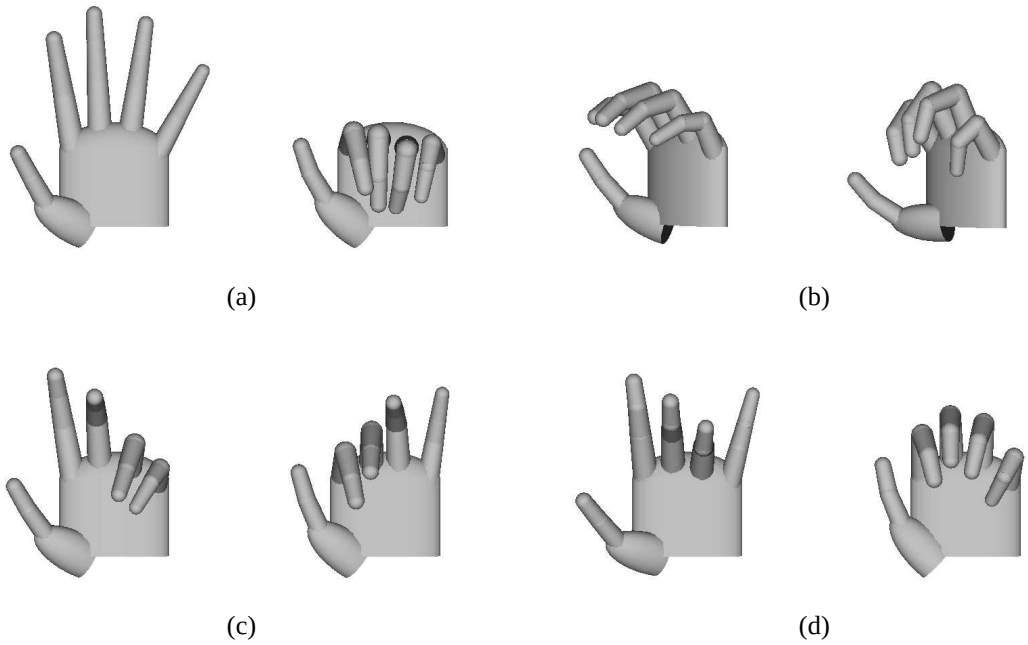


Figure 12: **Variation along principal components**. *This figures shows the variation in hand pose when moving away from the mean pose into the direction of the* **(a)** *first,* **(b)** *second,* **(c)** *third, and* **(d)** *fourth principal component. The input data set contained 50,000 joint angle vectors, obtained from a data glove. The subject was moving the fingers in a random way while trying to cover the possible range of motion.*

3.3 Constructing a tree for articulated motion

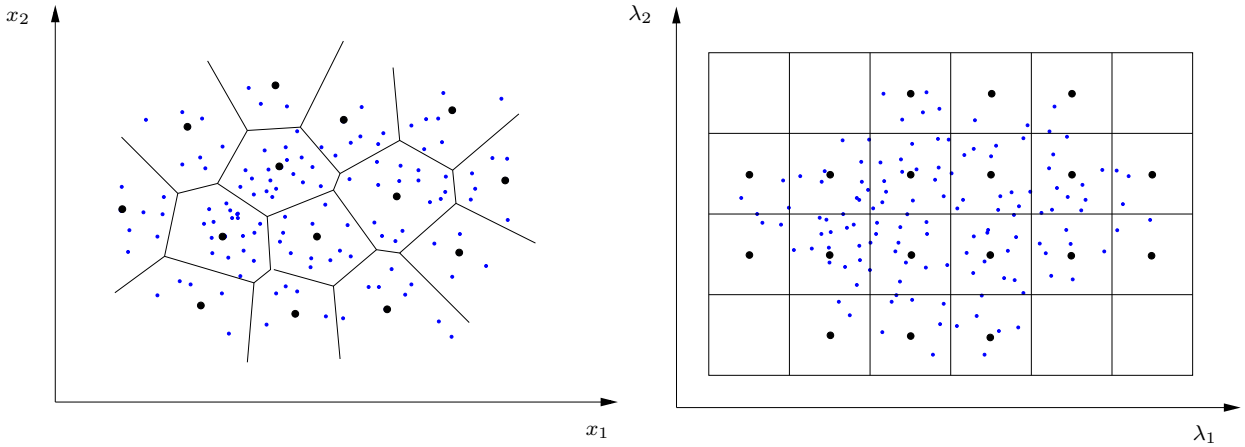
As mentioned above, finger joint angles are highly correlated. Even though the model has 21 DOF for finger articulation, it has been observed that less parameters are usually sufficient to

model articulated hand motion. Using a data glove, 15 sets of joint angles (sizes of data sets: 3000 to 264,000) were captured from three different subjects carrying out random hand gestures, trying to cover the possible range of finger motion. It was found in all cases that 95 percent of the variance was captured by the first eight principal components, in ten of the data sets within the first seven, which largely confirms the results reported in [45] on a larger data set. The variation along the first four principal components is illustrated in figure 12. Two methods to obtain the discrete sets $\{\mathcal{S}^{i,l}\}_{i=1}^{N_l},\ l = 1, \ldots, L$, have been implemented:

- **Clustering in parameter space:** Since the joint angle data lies within a compact region in state space, the data points can simply be clustered using a hierarchical k-means algorithm with the distance measure $d(\mathbf{x}_1, \mathbf{x}_2) = ||(\mathbf{x}1 - \mathbf{x}2) \mod \pi||_2$.

  A partition of the state space is given by the Voronoi diagram defined by these nodes, see figure 13a.

- **Partitioning the eigenspace:** The joint angle data is projected onto the first k principal components ($k < 21$), and the partitioning is done in the transformed parameter space. The centres of these regions are then used as nodes on one level. Only partitions, which contain data points need to be considered, see figure 13b. Multiple resolutions are obtained by subdividing each partition.



(a) (b)

Figure 13: **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.*

Both techniques are used to build a tree for tracking finger articulation. In the first sequence the subject alternates between four different gestures (see figure 14). For learning the transition distributions, a data set of size 7200 was captured while performing the four gestures a number of times in random order. In this experiment the tree is built by hierarchical k-means clustering of the whole training set. The tree has a depth of three, where the first level contains 300 templates

together with a partitioning of the translation space at a 20×20 pixel resolution. The second and third level each contain 7200 templates (i.e. the whole data set) and a translation search at 5×5 and 2×2 pixel resolution, respectively. The transition matrices $\mathbf{M}^{Ll}$ are obtained by histogramming the data [18, 43]. The tracking results for the tree constructed by clustering are shown in figure 14. No global parameters other than translation parallel to the image plane was estimated in this experiment. As before, initialization is handled automatically.

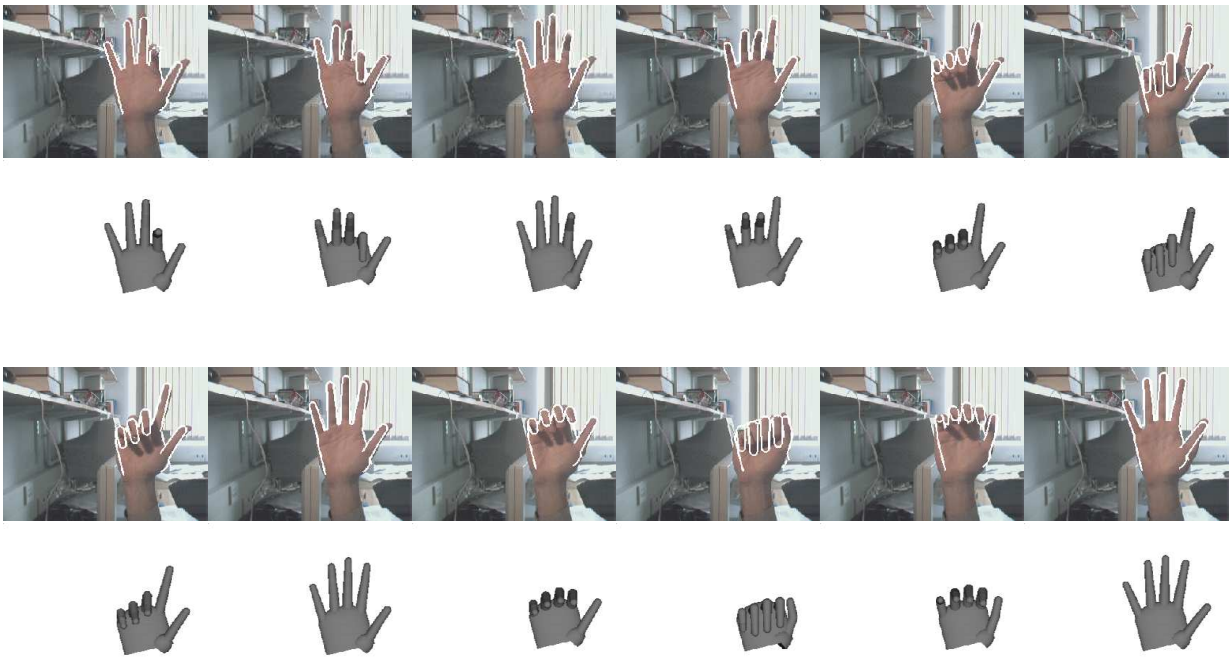


Figure 14: **Tracking articulated hand motion.** *In this sequence a number of different finger motions are tracked. The images are shown with projected contours superimposed (top) and corresponding 3D avatar models (bottom). The nodes in the tree are found by hierarchical clustering of training data in the parameter space. Dynamics are encoded as transition probabilities between the clusters.*

Figure 15 shows the results of tracking global hand motion together with finger articulation. In this case the opening and closing of the hand is captured by the first two eigenvectors, thus only two articulation parameters are estimated. For this sequence the range of global hand motion is restricted to a smaller region, but it still has 6 DOF. In total 35,000 templates are used at the leaf level. The resolution of the translation parameters is 20 pixels at the first level, 5 pixels on the second level, and 2 pixels at the leaf level. The out-of-image-plane rotation and the finger articulation are tracked successfully in this sequence. The RMS error for this sequence, measured as localization error against labelled ground truth, is shown in figure 9d. The mean RMS error is 9.3 pixels, this larger error compared to the other sequences is mainly attributable to the discretization error introduced by using only two parameters to model articulated motion. The execution time for this sequence is about three seconds per frame on a 2.4 GHz P4 computer.

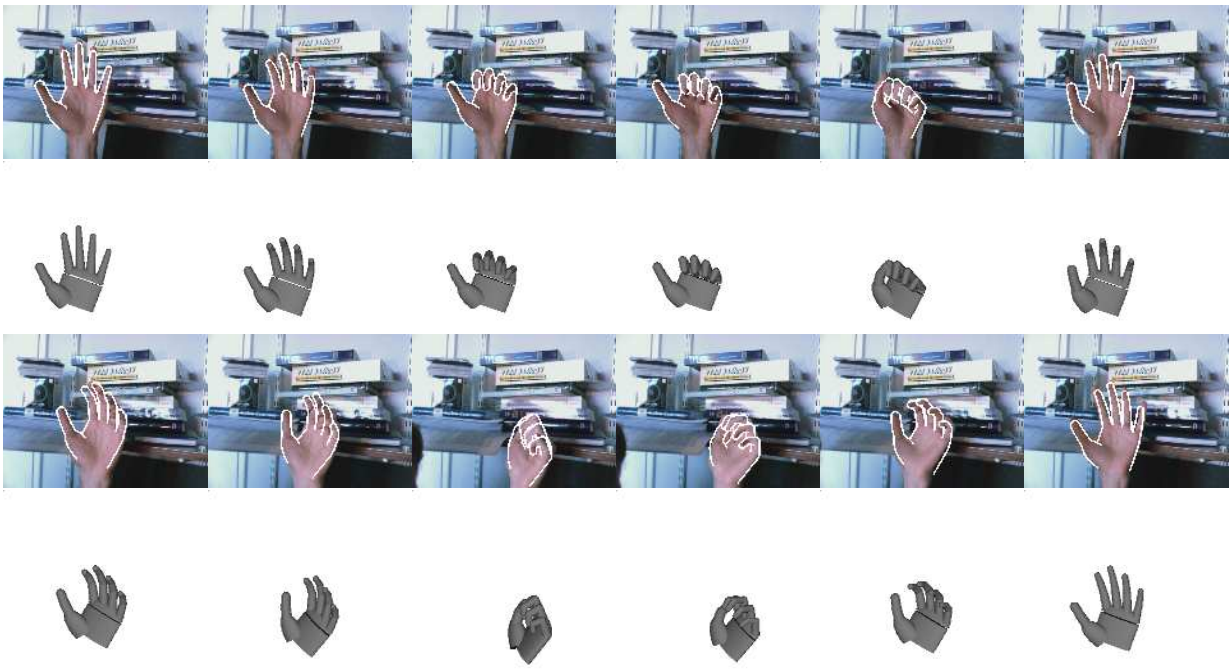


Figure 15: **Tracking a hand opening and closing with rigid body motion.** *This sequence is challenging because the hand undergoes translation and rotation while opening and closing the fingers. 6* DOF *for rigid body motion plus 2* DOF *for finger flexion and extension are tracked successfully.*

The purpose of the tree structure is efficiency, thus it is interesting to examine ways of constructing an optimal tree in terms of run-time. One approach to analyzing the performance of the method for the detection case is to interpret the template hierarchy as a tree of classifiers [3, 40]. A classifier $C^{i,l}$ at each node decides whether or not the current observation is within the region below that node. The aim is to design classifiers with high detection rates with modest false positive rates, minimizing the computational cost at the following levels. The expected run time for a tree below the node i at level l is given, similar to [3], by the recursion

$$E[T(C^{i,l})] = T^c(C^{i,l}) + E[pos(C^{i,l})] \sum_{j \in succ(i)} E[T(C^{j,l+1})], \tag{22}$$

where $T^c(C)$ is the run time of classifier C, $pos(C)$ is the detection rate of classifier C, and $succ(i)$ are the successors of node i. Minimizing this function requires simultaneous optimization of the tree structure as well as the threshold values.

The view of the nodes as classifiers also raises the question whether chamfer or silhouette template matching are optimal for classification [40]. It has been shown that classifiers trained with large sets of real image data perform better, however there is a trade-off between computation time and classification performance as shown in figure **??**. When used in a classification hierarchy, the detection rate of a classifier needs to be very high, so as not to miss any true positives. The false positive rate for each single pose classifier at a fixed detection rate of 0.99, is given in the last column of table **??**. Chamfer and Hausdorff matching, while having a larger false

positive rate, are about 10-14 times faster to evaluate than marginalized templates [40] and about 40 times faster than the trained linear classifier [14]. In addition, they only require the contour points to be stored in memory.

Classification Method	Number of Points	Execution Time	fp at $tp = 0.99$
Chamfer	400	13 ms	0.10
Hausdorff	400	13 ms	0.12
Marginalized template [40]	5,800	186 ms	0.02
Linear classifier [14]	16,384	524 ms	0.01

Figure 16: **Computation times for template correlation.** *The execution times for computing the dot product of 10,000 image patches of size* 128×128*, where only the non-zero coefficients are correlated for efficiency. The time for computing a distance transform or dilation, which needs to be only computed once for each frame when chamfer or Hausdorff matching is used, is less than 2 ms and is therefore negligible when matching a large number of templates. The last column shows the false positive (*fp*) rates for each single pose classifier at a fixed true positive (*tp*) rate of 0.99.*

4 Conclusions and future work

In this paper a framework was developed for tracking articulated hand motion from a video. The ability of the proposed hierarchical filter to recover 3D motion, even under self-occlusion, was demonstrated. The combination of hierarchical detection and Bayesian filtering has a number of benefits, in particular it specifically addresses the problems of initialization and recovery. The results in this paper were obtained using sequential on-line processing, motivated by applications in the HCI domain. However, the leaf nodes can be viewed as defining a discrete state model, thus it is straightforward to process the sequence using a batch algorithm as in hidden Markov models to optimize over the whole sequence [8, 24]. It should be noted that in contrast to particle filters [12, 21, 17] the tree-based filter is a deterministic filter, which was motivated by a number of problem-specific reasons. First of all, the facts that hand motion can be fast and that the hand can enter and leave the camera view in HCI applications call for a method that provides automatic initialization. The second important consideration is the fact that the time required for projecting the model is approximately three orders of magnitude higher than evaluating the likelihood function. This makes the sampling step in a particle filter costly and the off-line generation of templates attractive. The ideas of tree-based filtering and particle filtering can also be combined by using the posterior distribution estimated with the tree-based filter as an importance distribution for a particle filter in a way similar to [30]. The range of allowed hand motion is currently limited by the number of templates that need to be stored. This currently poses a main obstacle for extending this method to a full range hand tracker. There is a trade-off between generating templates online and storing them offline for fast access, i.e. memory usage vs. speed. The proposed method is to be used when speed is at a premium and memory is not. With better hardware acceleration model projection will be faster, while at the same time memory will also increase. One can expect, for example, that current systems using a PC cluster to store

large data bases of templates [35] will become much more compact. In fact, storing one million shape templates requires about 2 gigabytes, and keeping them in memory is a viable option on current PCs. Additionally faster online generation of templates as well as learning a relevant set of representative templates will allow a reduction of the memory requirement.

Acknowledgments

This work was supported by the EPSRC, the Gottlieb Daimler–and Karl Benz–Foundation, Toshiba Research, the Gates Cambridge Trust, and the ORS Programme.

References

[1] 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, DC, May 2002.

[2] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 432–439, Madison, WI, June 2003.

[3] S. Baker and S. K. Nayar. Pattern rejection. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 544–549, San Francisco, CA, June 1996.

[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] A. Blake and M. Isard. *Active Contours : The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, London, 1998.

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

[8] M. Brand. Shadow puppetry. In *Proc. 7th Int. Conf. on Computer Vision*, volume II, pages 1237–1244, Corfu, Greece, September 1999.

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

[10] 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.

[11] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 126–133, Hilton Head, SC, June 2000.

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

[13] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. A review on vision-based full DOF hand motion estimation. In *Proc. Workshop on Vision for Human-Computer Interaction*, San Diego, CA, June 2005.

[14] P. F. Felzenszwalb. Learning models for object recognition. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 56–62, Kauai, HI, December 2001.

[15] 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.

[16] 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.

[17] 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.

[18] A. J. Heap and D. C. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. 6th Int. Conf. on Computer Vision*, pages 344–349, Bombay, India, January 1998.

[19] 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, May 1993.

[20] 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.

[21] M. Isard and A. Blake. CONDENSATION — conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, August 1998.

[22] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *Proc. 6th Int. Conf. on Computer Vision*, pages 107–112, Bombay, India, January 1998.

[23] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

[24] N. Jojic, N. Petrovic, B. J. Frey, and T. S. Huang. Transformed hidden markov models: Estimating mixture models and inferring spatial transformations in video sequences. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume II, pages 26–33, Hilton Head, SC, June 2000.

[25] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances. *IEEE Trans. Automatic Control*, 45(3):477–482, March 2000.

[26] R. Lockton and A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *Proc. British Machine Vision Conference*, volume II, pages 817–826, Cardiff, UK, September 2002.

[27] 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.

[28] R. Navaratnam, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Hierarchical part-based human body pose estimation. In *Proc. British Machine Vision Conference*, volume 1, pages 479–488, Oxford, UK, September 2005.

[29] R. Okada, B. Stenger, T. Ike, and N. Kondoh. Virtual fashion show using marker-less motion capture. In *Proc. 7th Asian Conference on Computer Vision*, pages 801–810, Hyderabad, India, January 2006.

[30] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. 8th European Conf. on Computer Vision*, volume I, pages 28–39, Prague, Czech Republic, May 2004.

[31] 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.

[32] 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.

[33] R. Rosales, V. Athitsos, L. Sigal, and S. Scarloff. 3D hand pose reconstruction using specialized mappings. In *Proc. 8th Int. Conf. on Computer Vision*, volume I, pages 378–385, Vancouver, Canada, July 2001.

[34] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 750–757, Nice, France, October 2003.

[35] 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.

[36] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Proc. 6th European Conf. on Computer Vision*, volume 2, pages 702–718, 2000.

[37] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *Int. Journal of Robotics Research*, 22(6):371–393, 2003.

[38] H. W. Sorenson. *Bayesian Analysis of Time Series and Dynamic Models*, chapter Recursive Estimation for nonlinear dynamic systems, pages 127–165. Marcel Dekker inc., 1988.

[39] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based hand tracking using an unscented Kalman filter. In *Proc. British Machine Vision Conference*, volume I, pages 63–72, Manchester, UK, September 2001.

[40] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Hand pose estimation using hierarchical detection. In *Intl. Workshop on Human-Computer Interaction*, pages 105–116, Prague, Czech Republic, May 2004.

[41] A. Thayananthan, R. Navaratnam, P. H. S. Torr, and R. Cipolla. Likelihood models for template matching using the pdf projection theorem. In *Proc. British Machine Vision Conference*, pages 949–958, London, UK, September 2004.

[42] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proc. British Machine Vision Conference*, volume 2, pages 589–598, Norwich, UK, September 2003.

[43] K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int. Journal of Computer Vision*, 48(1):9–19, June 2002.

[44] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 511–518, Kauai, HI, December 2001.

[45] 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.

[46] H. Zhou and T. S. Huang. Tracking articulated hand motion with eigen-dynamics analysis. In *Proc. 9th Int. Conf. on Computer Vision*, volume II, pages 1102–1109, Nice, France, October 2003.

Björn Stenger received the Dipl.-Inf. degree from the University of Bonn, Germany, in 2000 and the PhD degree from the University of Cambridge, UK, in 2004. His thesis on hand tracking won the BMVA Sullivan Thesis Prize. He is currently a research fellow at the Toshiba Corporate R&D Center in Kawasaki, Japan, where he is working on gesture interfaces and real-time body tracking.

Arasanathan Thayananthan received his MEng degree in Engineering and Computing Science from the University of Oxford in 2001. He completed his PhD degree at the University of Cambridge in 2005 and is currently working as a research associate at the Department of Engineering, University of Cambridge.

Philip H. S. Torr did his PhD (DPhil) at the Robotics Research Group of the University of Oxford under Professor David Murray of the Active Vision Group. He worked for another three years at Oxford as a research fellow. He left Oxford to work for six years as a research scientist for Microsoft Research, first in Redmond USA in the Vision Technology Group, then in Cambridge UK founding the vision side of the Machine learning and perception group. He is now a Professor in Computer Vision and Machine Learning at Oxford Brookes University. Philip Torr won the Marr prize in 1998. He was involved in the algorithm design for Boujou released by 2D3. Boujou has won a clutch of industry awards, including Computer Graphics World Innovation Award, IABM Peter Wayne Award, and CATS Award for Innovation, and an EMMY. He is a senior member of the IEEE and on the editorial boards of ACM Computers and Entertainment, IEEE Transactions on Pattern Analysis and Machine Intelligence and the Journal of Image and Vision Computing.

Roberto Cipolla received his first degree in Engineering from the University of Cambridge in 1984; an MSE degree (Electrical Engineering) from the University of Pennsylvania in 1985; and the MEng degree from the University of Electro-communications in Tokyo in 1988. In 1991 he was awarded a D.Phil. in Computer Vision from the University of Oxford. Since 1992 he has been at the Department of Engineering in the University of Cambridge where he is now Professor of Information Engineering. His research interests are in computer vision and robotics and include the recovery of motion and 3D shape of visible surfaces from image sequences; object detection and recognition; visual tracking; robot hand-eye coordination and novel man-machine interfaces using visual gestures. He has authored 3 books, edited 6 volumes and co-authored more than 250 papers.