2.1 Introduction

Research on vision-based sensing of hands was first reported in the early 1980s, but the last decade has seen a burgeoning of the field, driven, of course, by cumulative progress in vision algorithms, but also by advances in computing and camera hardware and the perceived value of potential applications. This chapter presents a detailed survey of hand pose estimation and tracking methods. A number of methods for human motion capture have also been included, because of their parallels with hand tracking.

The review has been divided in three sections. Section 2.2 is concerned with 2D methods, and starts by reviewing image-based methods that run “model-free” in Section 2.2.1, methods based on exemplars are reviewed in Section 2.2.2, and those that run with with 2D models are considered in Section 2.2.3.

Section 2.3 is concerned with those methods where the model is built in 3D. There is a considerable number of methods that use partial 3D models, and these are outlined in Section 2.3.1, 2.3.2 and 2.3.3. Methods that include quite complete kinematic models are less varied, and these are examined in Section 2.3.4, 2.3.7.

Section 2.4 of the review considers the intermediate class of methods which has come to prominence in recent years where 3D shapes are deduced directly from image appearance. High-level methods for action and intention recognition are not included, and little attention is given to gesture recognition meth-
ods. The survey concludes with a summary, a list of challenges that still remain, and some considerations as to likely future directions in the field.

## 2.2 2D methods

Two dimensional or image-based approaches are applicable to problems where it is enough to recover a two-dimensional description of the hand pose and a qualitative description of the gesture. The number of recognisable gestures is limited, but they can be very robust.

### 2.2.1 Model-free methods

Some methods do not model the hand’s appearance. Instead they track a cloud of moving features or blobs that are likely to be hands, and attach meaning to them [FAB+98, Que95, WADP97, BPH98, WLH00]. One of the most robust model-free methods is that of Kölsh and Turk [KT04b]. It uses a set of KLT feature trackers (named after Kanade, Lucas [LK81] and Tomasi [ST94]) which is initialised in the hand region (see Figure 2.1). The position of these features is bounded by the skin colour blob and a geometric constraint that prevents features being located too close to or too far from each other. The hand position is determined by the median feature among the set. This system is able to work at video rate, with performance superior both to that of a raw KLT tracker and to that of a mean-shift tracker [Bra98].

![Figure 2.1: 200×230 pixel areas cropped from the 720×480-sized frames of a video sequence showing the tracking result with highly articulated hand motion. The cloud of dots represents the flock of features, the large dot is their mean.](From KT04b, reproduced with permission.)

### 2.2.2 Exemplar-based methods

Exemplar-based representations are simple to create and are capable of representing highly nonlinear configuration manifolds. Exemplars can be thought of as minimally preprocessed select representatives of the training data itself, which together ‘span’ the range of the modelled entity [And01]. Some
exemplar-based methods do not present any pre-processing step and use whole image patches to represent instances of the target object. For instance, Darrell and Penland \cite{DP95} simply use correlation for gesture recognition. In some cases, even parts of the background are included. For this reason, such methods are usually memory-intensive. In most cases, accuracy and speed are achieved by using fast search methods, powerful classifiers and temporal information.

**Efficient classification methods**

Gavrila and Philomin \cite{GP99} use chamfer matching and a coarse-to-fine search in the image grid to speed up matching. In order to detect multiple body poses, a database of shape templates is partitioned into a number of clusters following their dissimilarity. Clustering is done recursively, leading to the creation of a tree of templates. Matching is then done by traversing the tree structures, from root to leaves, following the path with most similarity with the observed image. This prunes a large number of comparisons that would be done if exhaustive search was used. Pedestrian detection is performed at near video-rate, but the accuracy of this system is between 75-85% of detection rate.

Boosting \cite{RO00} is a fast and powerful classification technique based on a weighted cascade of weak but quick classifiers. This combination provides accurate results with a good description of the decision boundary. Viola and Jones \cite{VJ01} proposed a method for object recognition based on boosting of simple image features founded on Haar wavelets. Their results for face detection make this a gold standard method for this task. Although its application is quick, the training phase is computationally demanding. Variations upon the method have been applied to hand tracking, with nuances to improve the training time \cite{KT04a}, \cite{JRM06}, \cite{WAP06}. Lockton and Fitzgibbon \cite{LF02} applied this for real-time gesture recognition to replace keyboard and mouse.

**Using temporal information**

A variety of techniques has been employed to model temporal sequences, including PNF (past, now, future) networks \cite{PB98}, tree-based search \cite{Lyo02}, finite state machines \cite{HTH00} and, most commonly, Hidden Markov Models (HMMs) \cite{WBAS03}, probably due to its success in speech recognition.

Starner et al. \cite{SWP98a} use HMM to represent a lexicon with four states to recognise some gestures of American sign language (ASL). The silhouette of the hands are obtained with skin detection and they
are described using a 16 element feature vector built from basic moment-like blob measurements. In [TB02], Toyama and Blake show how two dynamic processes (global motion and shape changes) can share the same joint observation density provided by the chamfer distance. This leads to an attractive homogeneity of description and implementation. The drawback is that this requires the use of a large particle set, which must simultaneously represent hypotheses of both shape and position. Fei and Reid [FR04] argue that in many applications these processes can (or even should) be decoupled, potentially leading to a more economical use of particles and hence to greater efficiency and reliability. They propose a method for the analysis of complex hand motion that assumes that the hand motion consists of two independent components: cyclic shape variations and hand region motion. The former is modelled by an HMM using silhouette moments, the latter is a particle-based colour region tracker. Each maintains a full PDF of its respective component of the motion, and they interact via an importance sampling function.

Matching image information

This review turns to consider how the observed image is matched with prior knowledge of the hand’s appearance. The simplest method in terms of implementation is template matching or normalised cross correlation as used in [FAB+98]; but it is very common to use shape descriptors [dCJ01] for hand pose classification [YI98], image moments [Hu62] being popular descriptors for this task [FAB+98, Lyo02]. Other descriptors employed for matching are (i) those based on the analysis of the curvature of the silhouette contour [HSS02] and (ii) polar histograms [OS03a] computed from the centre of the silhouette of the hand [WAP06, FTR+04]. (The second reference uses a multiple offset flash gun to help recover depth edges in hand images. The illustration and results in Figure 2.2 are a good example of the difficulty
caused by partial occlusion.) In [TSTC03], Thayananthan et al. compare two methods for matching of Canny edge images. One method is based on shape context and the other combines an exemplar-based detector based on chamfer matching with an optimisation method for model-fitting. They conclude that chamfer matching outperforms shape contexts, but chamfer matching requires a high number of templates for matching.

### 2.2.3 Trackers with 2D object models

Prior knowledge can be exploited to obtain a more quantitative description of the hand shape. Model-based methods use a description of possible hand shapes and tracking is performed by matching the model pose with the observed hand images.

Active contours [BI98] as well as deformable templates [BH95, HCT95, BS02, Bow99, KH95, BF95, TLC+98] have been used to model the hand’s appearance in images, methods robust to small variations in pose and shape of the hand. Deformable 2D templates comprise points on the outline of the hand that are used as interpolation nodes for an approximation of the outline. The template sets and their corresponding variability parameters are stored, and matching involves minimising the summed squares of differences between points from the image silhouette and the templates. This is a simple and successful method if the original viewpoint is maintained, or if characteristic views are available. Triesch and von der Malsburg [Tv01] employed a variation on this approach, elastic graph matching, in which the hand image is represented as a labelled graph. The distribution of nodes describes postures in 2D.

Another image-based method to track an articulated model was proposed by MacCormick and Isard [MI00]. With an articulated model and active contours to segment the index finger, they tracked four degrees of freedom, namely planar translation, orientation of the thumb and of the index finger, using CONDENSATION [IB98, IB96]. Unlike model-free and classification-based methods, this system recovers continuous parameters rather than recognising gestures from a discrete “vocabulary”. In [MI00], the authors show the application of this tracker to implement a virtual workspace with a more natural interface for drawing objects, as shown in Figure 2.3.

For whole body tracking, more complex articulated models have been used recently assuming that each limb part can be modelled in the image by a rectangular shape. Bregler et al. [BOC+98] model the body segments with a multi-dimensional mixture of Gaussian blobs, modelling motion, shape and grey
level distributions. The blobs are initialised using motion coherence likelihoods, based on optical flow. For a two parts articulated body, they use Expectation-Maximisation to estimate the pose with simple kinematic priors to constrain the blob estimation. In their “cardboard people” tracker Ju et al. [JBY96b] use three different views, viz frontal, oblique and side on, to design two-dimensional templates that represent projections of the object in each view. They track walking motion, but it is assumed that the orientation of the object does not change along the sequence. Lu et al. [LPV06] use a planar layered model capable of handling occlusion for gait analysis. For each body part, it actively selects which side is more likely to have reliable edges for walking movements (avoiding, for example, edges between the legs because they are often perturbed by clothing). Local tracking of body parts is based on mean-shift, and strong motion priors (such as arms move in opposition to thighs) are included. Good results are reported for examples which conform with these priors.

In [MR98] Morris and Rehg model the projected motion of an inter-joint link in the scene as affine flow patches with imposed kinematic constraints. This is similar to Ju’s model [JBY96b], but with fewer parameters and a more direct connection to the underlying articulated motion. These two approaches are compared in Chapter 6.

The above methods give continuous pose estimates in 2D, which is not always required in problems related to registration of articulated objects. Felzenszwalb and Huttenlocher [FH00, FH04] proposed the use of pictorial structures for object recognition, based on dynamic programming with discretisation of the parameter space. The algorithm searches the parameter space to minimise a cost function that
2.2 2D methods

combines the matching score of the object parts with kinematic constraints. The matching score is based on how well rectangles can fit areas segmented by background subtraction. The joints have a spring-like model and the kinematic constraints try to minimise their degree of distortion from an up-right pose.

Ronfard et al. [RST02] built a similar system, but they replace the rather simple part detectors with dedicated detectors learned for each body part using Relevance Vector Machines (RVMs) [Tip01], which are support vector machines-like classifiers that offer a well-founded probabilistic interpretation and improved sparsity for reduced computation.

Ramanan and Forsyth [RF03] use clustering to learn the appearance of objects that move in a video sequence. This approach, called foreground enhancement, is different from traditional background subtraction since it is used to learn the appearance and not to find people. Therefore, once the appearance is known, they can track people who initially stand still, so long as they move at some point. They use a probabilistic graphical model to locate and track multiple people in video sequences.

Kumar et al. [KTZ04] extend Felzenszwalb and Huttenlocher’s approach [FH04] by using a complete graph model, rather than a tree structure. In order to estimate the maximum a posteriori estimate of the pose and shape parameters, a loopy belief propagation method is used, which is a message passing Viterbi-like algorithm for graphs with loops. The authors show that this gives more constraints for the pose estimation and better results than a tree structure.

More robust to unconventional human body poses is the method of Mori et al. [MREM04]. It uses a method for segmentation that gives the probability boundaries based on brightness and texture. This is applied with two different parameters: one that segments regions of the image large enough to be likely to contain half limbs and torso segments, and one that super-segments the image, giving super-pixels. Next, a method for detection of salient body parts is applied to the large segments. This method is based on four cues: contour, shape, shading and focus. These cues are combined and the regions with highest score are selected and combined using constraints on relative widths, lengths, adjacency, and similarity of clothing. The output of this system is a ranked shortlist of possible configurations of the human body in the image. Each pose configuration is obtained from the association of different segments of the images, which enables the computation of a body segmentation (see Figure 2.4). The main drawback of this system is the dependence on its training set for robustness. Furthermore, the design of this method
was driven by its data: baseball players images. This allowed the use of specific features, such as the symmetry and regularity of the players’ uniforms, which, in most of the showed images, are highly distinguishable from the background.

![Diagram showing the process of 2D methods](image)

Figure 2.4: (a): Data flow the algorithm of Mori et al. [MREM04]. (b-e): Selected result from the short-list of final configurations: (b) input image, (c) candidate half-limbs, (d) extracted body configuration, (e) associated segmentation. (© [MREM04], reproduced with permission.)

The method above estimates pose from single images independently. This review now turns to methods that exploit spatio-temporal information.

Yacoob and Davis [YD98] use an approach for learning and estimating temporal flow models from image sequences. Such models are created by applying principal component analysis to time sequences of parametric models of body part motion. These observations are obtained using the “cardboard body” of [JBY96a]. This approach bridges the gap between traditional instantaneous optical flow estimation and multi-frame motion estimation. The learned motion models are used in a spatio-temporally constrained image-motion formulation for simultaneous estimation of several rigid and non-rigid motions.

Wu et al. [WHY03] also explore temporal priors, but instead of constraining a spatio-temporal manifold, a motion filter is used. Articulated objects are not modelled as single objects with low-DOF joints. Instead, each rigid part \( k \) is measured \( (z_k) \) in the image independently and the pose of each part is redundantly described by its pose \( x_k \), independent from the other parts of the object. The measurements of each part give a local likelihood \( p_k(z_k|x_k) \) for the pose. The local prior \( p_k(z_k) \) can be obtained using temporal information, thus it depends on previous measurements. This is combined with neighbourhood
priors, which constrain this part to be connected to its neighbours. The system is modelled with a dynamic Markov network, which serves as a generative model for the articulated motion. In their most challenging experiment, a 10-part articulated body was tracked at 0.56 frames/second, using 200 particles per part, but the lack of different texture on hand parts makes this difficult for hands. Figure 2.5 illustrates the results for a 3 parts finger.

2.2.4 Discussion

2D image-based methods simplify tracking and pose estimation by focusing on motions that are parallel to the camera plane and by restricting the appearance changes. A considerable robustness is achieved with methods that model (or are invariant to) scale and shape changes, but they either do not provide enough information about the hand pose (e.g. model-free methods) or they are view-dependent. For instance, methods that use image-based articulated models can struggle with fingers pointing at the camera. Such methods have not successfully been applied to full-DOF whole hand tracking because the lack of strong textures challenges their effectiveness.

2.3 3D model-based tracking

The models in the previous section were two dimensional. This review now considers methods based on 3D models, starting with those that employ a simple rigid model to track hands, moving on to methods in which specific image features locations are used, and ending with those that model many or all of the available degrees of freedom.

2.3.1 Tracking hands in 3D without estimating fingers joint angles

Several early 3D hand trackers used rigid hand models, with the intention of using the hand as a 3D pointer or mouse. In [COK93] Cipolla et al. tracked four coloured markers on a hand (three on the
ends of fingers) to recover 3D orientation which was then used to control the orientation of a graphics model. Without using colour markers, Cipolla and Hollinghurst [CH98] tracked the thumb and index finger using active contours. The intersection between the finger line visible from the two cameras and a ground plane was computed, allowing for simple interaction with a robot. Bretzner and Lindeberg [BL98] used three cameras to track three fingers to establish both the position and orientation of a rigid hand in 3D. An ingenious alternative to the using multiple views was the use in [SK99] and [SLM+02] of a single camera with multiple light sources to cast shadows onto a planar surface.

In [OZ00], O’Hagan and Zelinsky propose decoupling gesture recognition and pose estimation. Their system assumes a 3D rigid planar hand model. A curvature analysis allows features on the outline of the hand silhouette to be selected in each image of a stereo pair, allowing the planar pose of the hand palm to be determined. An image-based classification is combined with 3D rigid hand tracking to recognise gestures. More practical results are shown by Segen and Kumar [SK98, SK00] in a video-rate application. As in O’Hagan and Zelinsky, gestures are recognised using peak and valley detection on the hand silhouette. A finite state machine analyses movements to refine gesture recognition. The same image features are also used to estimate 3D position, azimuth and elevation for both the index finger and the thumb. This system is applied to 3D scene composition and navigation.

Sato et al. [SSK01] use skin detection to segment hand blobs in a two cameras system, allowing the 3D position to be computed through triangulation of the centre of the hand. The orientation is then determined using the principal axis of the hand and the left and right end points. A small set of gestures is recognised using a neural network applied to segmented, normalised and sub-sampled hand images.

### 2.3.2 3D tracking of an articulated arm with fixed basis

A number of authors have modelled and tracked hands as a rigid extensions of the forearm, articulated at the elbow and shoulder. Gonçalves et al. [GdUP95, BGP96] developed a monocular system capable of tracking human arm in 3D where the limbs are modelled as truncated cones, the shoulder is a spherical joint and the elbow is a planar joint, giving the model 4 DOF. The image measurements are obtained by thresholding and smoothing the image, and the method performs 1D searches for the highest gradient perpendicular to the projection of each limb segment (forearm, arm and hand tip). Only five control points per segment are searched. Tracking is performed by a recursive estimator that performs random
walk in the spherical joint velocities and uses the Extended Kalman Filter. Ambiguities are avoided by constraining angles. Tracking was achieved at 11 Hz (in 1995, when typical processor speeds were 100 MHz) and the standard deviation of the estimates of hand tip position is some 1% of the distance between the camera and the user’s hand.

Vogler and Metaxas [VM96] use three near-orthogonal views and impose priors on the human body shape. Deformable silhouettes are used and heuristics are applied to locate the position of joints. For tracking, the method actively selects the best viewpoint for each body part in each frame [KM96] and, once selected, planar rotations and translations are estimated to drive updates to the 3D model. A Kalman filter is used for prediction and gesture recognition is performed using an HMM to recognise a set of 53 gestures from ASL. (The word accuracy achieved was of 88% for the 3D context-dependent experiments with 456 testing gestures. The authors made use of a commercial HMC system based on magnets interchangeably with their vision-based method.)

2.3.3 Using finger tip locations

If a reliable estimation of the position of the finger tips is available, it is possible to obtain solutions for hand pose using inverse kinematics [Cra89]. Both single [CGH02], [LH00] and multiple cameras [Lie04], [Reh95] have been used (as, incidentally, has active illumination [SKK00] and laser tracking [PCI03]), and a wide variety of methods have been proposed to detect and locate fingertips. These include (i) coloured markers [Lie04, CGH02, LH00, RL00]; (ii) circle detection by fitting [vHB01] and Hough transforms [CC03]; (iii) line detection with the Hough transform [Ahm95, GW00]; (iv) curvature analysis [YI98]; (v) correlation [Reh95]; and (vi) trained neural nets [NR99]. To eliminate ambiguities constraints imposed by limitations of muscles and tendons must be included in the model. Most use hard-coded linear dynamic constraints between joint angles to reduce the dimensionality [LH98, CGH02].

Motion priors are used to predict over periods of occlusion [LH00, Lie04, RL00].

Perhaps the main advantage of these methods is that they do not require a model of the hand to be back-projected into the image. However, if no colour markers are used, detection of fingertips is very challenging particularly if the image region around a fingertip is skin coloured, as is common situation

\[^1\]For instance, the relation between the proximal inter-phalangeal \(\theta_1\) and the distal inter-phalangeal \(\theta_2\) joint angles is modelled as \(\theta_1 = \frac{2}{3}\theta_2\). The abduction dofs are often ignored as in [CGH02].
when the fingers are bent. Even when markers are used, the measurement of the palm is made unreliable by skin movement.

### 2.3.4 Marker-less articulated tracking using complete 3D models

The methods described on this section onwards involve the use of a complete 3D model.

#### Symmetric tracking with a kinematic chain

In the early 1990s Rehg and Kanade [RK94, Reh95] developed the first system to track unmarked hands using a realistic (27 DOF) 3D kinematic chain at near video rates. Finger phalanges were modelled as simple cylinders, fingertips as halves of spheres, and the palm as a couple of planes linking two cylinders.

Two feature extractors to measure the sum of squared differences (SSD) were presented: deformable templates registration and point and line features. In template registration, the cost function is based on intensity errors used to measure the geometric misalignment between an input image and the image predicted by the projected kinematic model. Each finger is described by a planar template deformed with an affine transform to approximate the projection. Templates provide a useful level of generality, and make it possible to exploit arbitrary texture cues. But for a specific object like the hand, the constraints provided by the template matching can be approximated by purely geometric error functions involving point and line features [RK94].

Point and line features tracking is performed by projecting the middle axes of the truncated cylinders onto the image and searching for edges in directions perpendicular to the projected segments. Search for edge in the finger tips is also performed. The significantly lower computational cost of computing point and line features makes on-line tracking possible. The residual error between the estimated position of the features and the actual located features are combined and minimised using a weighted Gauss-Newton iterative method to estimate the state update as follows:

\[
q_{k+1} = q_k - \left[ J_k^T J_k + S \right]^{-1} J_k^T R_k,
\]

(2.1)

where \( J_k \) is the Jacobian matrix for the residual \( R_k \), both of which are evaluated with the state vector \( q_k \) and \( k \) is the iteration index. \( S \) is a constant diagonal conditioning matrix used to stabilise the least squares solution in the presence of kinematic singularities.
2.3 3D model-based tracking

The method was implemented on multiple processors using separate frame grabbers for the two cameras and a separate computer to render and display the estimated model, resulting in a 10 Hz tracking of 19 degrees of freedom (where the middle fingers were not tracked) and 7 Hz on all 27 degrees of freedom. The on-line version did not include Rehg’s method of occlusion reasoning, which was restricted to off-line because of its computational complexity. Although the palm is modelled, for simplicity its projection is not used for tracking.

Lu et al. [LMSO03] describe a method for hand tracking using a single view from a motion sequence. A combination of spheres and truncated cones models the appearance of each part of the hand. Three image cues are used for tracking: edges, optical flow and shading information. Since there is not much image features on bare hands, the standard optical flow could not provide good results. As a solution, optical flow and shading information are combined using a generalised version of the gradient-based optical flow constraint that includes shading flow, i.e., the variation of the shading of the object as it rotates with respect to the light source. Similarly to Rehg’s work, 2D image feature discrepancies drive changes in 3D pose via the Jacobian. To combine the multiple cues, Lu et al. use Lagrange multipliers. An iterative method to impose joint constraints is also described. Basically, once the pose estimation results on some of the joints moving further than its limit, the joint is fixed to its limit and a new solution is estimated with this joint modelled as a rigid object. Tracking at 4 Hz was achieved on a Pentium 4 1 GHz cpu.

An alternative to Rehg and Kanade’s notation was proposed by Bregler and Malik in [BM98]. Instead of using standard full projective geometry, scaled orthography is used. Thus the effects of changes in distance from the camera are compensated by changes in scale of the object. This seem to be appropriate for problems with unknown camera calibration and objects far from the camera, as it is common for full body tracking. The image measurements are based on comparisons of internal pixels of warped image of object parts. The method is formulated using Lie Algebra, i.e., the motion between each pair of object parts is represented using a combination of the exponential of the canonical matrix of the coordinate frame of each DOF. These are used to build a linear relationship between instantaneous motion and pose change, allowing to obtain a least squares approximation of the pose update (exponential twist) for articulated objects, given the image measurements. As shown in Chapter 5, this turns out to equivalent to
the motion screw obtained by standard Jacobian-based methods for articulated object pose update. The experiments in [BM98] show successful tracking of a 6 DOF human body using a single camera and of a 19 DOF body using three cameras and a well-known “Eadweard Muybridge” sequence. The frame-rate achieved is not reported.

**Whole body tracking using a quantised feature space**

In their influential work, Gavrila and Davis [GD96] modelled the human body using superquadrics. Four widely spaced cameras were used, and the model projected into each of them under perspective. A fitting cost was defined by chamfer matching in a filtered and background-subtracted edge image, and coarse-to-fine search in parameter space used to determine the best-fitting quadrics.

A local best-first search was used for pose update. However, using 22 dimensions per human makes the search space large, and brute-force search daunting. Instead they proposed a search space decomposition in which the parameter space was recursively partitioned in a tree-like structure of subsets of parameters. At the leaf level were single parameters that were optimised individually, but the whole parameter set was used to verify the error. Once a parameter was optimised, it was fixed while the remainder were optimised. The parameters that have not been searched yet keep the predicted values from the previous iteration. This is an asymmetric search method, and different results can be obtained if different orders are used. The authors’ preferred order was persons; followed by head/torso position and inclination; torso twist; then arm pose.

The method worked well provided the image conditions were made benign — the subjects wore tight-fitting clothes of contrasting colour and the motion was straightforward. Tracking more complex motions, such as their tango sequence, required manual intervention.

### 2.3.5 Model refinement

In order to adapt the hand model to different users, some researchers have also optimised the “static” body parameters. Lu et al. [LMSO03] for example refine the length and thickness of fingers while tracking. During the first frames of a sequence, and after pose updates, the residual errors in edges and optical flow are accumulated and used to modify the hand shape by anisotropic scaling. Bregler and Malik [BMP04], show that the state space of their earlier motion tracking framework [BM98] can
be extended to also optimise over the kinematic model, and over the complete image sequence instead of just image pairs. The twist (state of the pose parameters) is kept fixed to the values obtained by the tracker and the equations are rearranged to optimise the length of each link of the articulated body. Based on Tomasi-Kanade’s factorisation method [TK92], Bregler’s system of equations is iteratively factored to optimise the articulated model and update the pose and shape parameters along a video sequence. A more advanced model is that of Plänkers and Fua [PF02] who refine meta-balls (generalised algebraic surfaces defined by a summation over \( n \) 3D Gaussian density distributions) attached to an articulated skeleton. The meta-balls simulate the gross behaviour of bone, muscles and fat tissue. The model is projected into the images and its silhouette is extracted. Tracking is performed in four steps (i) The silhouette of previous frame serves as initialisation for current frame; (ii) Optimise using active contours on disparity-filtered gradient image; (iii) Refine the body model to stereo data constrained by current silhouette estimate; and lastly (iv) Optimise the silhouette of the fitted model using active contours.

The use of a model that accurately reproduces the objects appearance must have a positive effect on tracking precision. However, there has been no analysis of whether, when resources are finite, such improvements compensate for the extra computational effort necessary for update and projection of a detailed model as it moves.

### 2.3.6 Motion filters

Motion filters have been used in many hand tracking methods to smooth the pose estimate and to provide predictions that improve the reliability of the tracker. Some have already been mentioned in passing.

#### Methods based on the Kalman filter

Shimada and Shirai [SS96] use the Extended Kalman filter (EKF) for monocular hand tracking in 3D and also allow model refinement by including the length of finger parts in the state vector. First, the best fitting solution is obtained with EKF and then this solution is modified applying inequality constraints based on human hand physiological restrictions. In cases where multiple solutions satisfy the constraints, multiple hypotheses are generated (based on symmetry w.r.t. the image plane) and their fitness is evaluated. This system was only evaluated using simulations. Wachter and Nagel’s persons tracker [WN99] uses an Iterative Extended Kalman Filter (IEKF) which consistently integrates edges and image texture cues for
the pose update.

Stenger et al. [SMC01] use an Unscented Kalman Filter (UKF) [JU97] to update the pose of their model, which like Rehg’s has 27 DOF, but it is built from 39 truncated quadrics (Figure 2.11), giving, of course conic projections. The hand dynamics are modelled using position, velocity and acceleration. The UKF is found to be more tolerant of non-linearities than the EKF, and permits higher frame rates than more sophisticated estimation methods such as particle filtering.

![Figure 2.6: (a) wire frame of the 3D hand model used by Stenger et al. [SMC01]. (b) Projection of this model on the input image during tracking.](copyright [SMC01]. reproduced with permission.)

**Stochastic and multiple hypotheses search strategies**

The challenges of unconstrained tracking and 3D tracking from monocular vision have lead to the research in methods to avoid local minima caused by ambiguities and configurations with singularities.

Deutscher et al. [DNBB99] have demonstrated that probability density functions (PDFs) for kinematic variables such as joint angles are actually non-Gaussian. This tends to happen particularly often in joint angle PDFs near their end-stop values and close to singularities where the kinematic chain lies in physically distinct but visually indistinguishable configurations. Their solution for human body tracking was to use **CONDENSATION**. Sidenbladh et al. [SBF00] presented a method similar to that of Deutscher’s, but they use limb texture in addition to edges alone.

Another stochastic solution was proposed for hand tracking by Nirei et al. [NSMO96]. Given a rough initial estimate obtained by mouse clicks, a Genetic Algorithm was used to minimise the estimation error of optical flow and maximise the overlap between the projected model and silhouette images using the chamfer distance. They then applied Simulated Annealing to refine the pose estimate. The results were
not obtained in real-time (unsurprisingly!), but they demonstrate that all the fingers could successfully be tracked in a short video sequence.

Stochastic tracking frameworks such as Condensation are capable of dealing with complex PDFs and avoid local minima, but the curse of dimensionality threatens this approach. The minimum number of particles required for successful tracking is exponentially proportional to the dimensionality of the problem [JDM00]. One of the issues that make Condensation computationally expensive is the definition and evaluation of the likelihood. Deutscher, Blake and Reid [DBR00] address this problem by developing the Annealed Particle Filter (APF) which uses a weighting function to approximate the likelihood. This weighting function is easy to be calculated and, unlike Condensation, the perturbation of the particles always decreases with time. This allows the use of much larger particle distributions with less computational effort. Davison, Deutscher and Reid [DDR01a] demonstrated the application of this algorithm for Human Motion Capture for character animation (see Figure 2.7). The particle distribution is used by the APF to evaluate several parameters of the weighting function in attempts to find a value that minimises it. Clever search strategies are needed to help particles to locate the global minimum of the weighting function to overcome the complexity of the search space. APF tends to be rather wasteful of computational resources in the searching of configuration space. At each time step, the APF must add a noise vector to the particles. The noise has to be large enough to lead to a search that covers a sufficiently large volume of the configuration space. This improves the tracking results, but many particles are wasted in randomly generated configurations.

![Figure 2.7: (a) Projection of the 3D body model on the image of one of the 3 view from a handstand video sequence used by Davison et al. [DDR01a]. (b) The virtual character on the pose obtained by the APF algorithm. The curve shows the trajectory of the base of the subject’s spine. (©[DDR01a], reproduced with permission.)](image)
To address this problem, Sminchisescu and Triggs [ST01b, ST01a] adopted a covariance weighted sampling in which a covariance matrix representing uncertainty is associated to each body pose hypothesis. This allows iterative generation of hypotheses that are less ambiguous, resulting in a more efficient distribution of the available tracking estimates. In parallel, the searching method of APF was improved by Deutscher et al. [DDR01b] by adding noise to each individual parameter of a particle in proportion to the variance observed in that parameter across the particle set. Another improvement was the use of a genetic algorithm-like particle crossover operator. This update on the algorithm lead to a 4-fold increase in processing speed.

An alternative bottom-up approach has been presented by Sigal et al. [SISB03]. They represent body parts individually and a stochastic algorithm places the parts randomly in 3D. A graphical model based on message passing and learning combines image measurements and spatial constraints and, in [SBR04], also temporal constraints. Bottom-up part detectors based on PCA of concatenated images (of multiple views) are used to aid detection of parts. They suggest their results improve on the APF because errors are not accumulated. However, although the optimisation searches for solutions that do not violate the constraints between body parts, these are not hard constraints and the system may provide impossible body configurations. (The stochastic method used is similar to that of [WHY03], mentioned earlier (page 19).)

Bray et al. [BKMM04] proposed the Stochastic Meta-Descent (SMD) method for hand tracking. It is a gradient descent method with local step size adaptation that combines rapid convergence with scalability and, as only a single hypothesis is considered, requires fewer samples than Condensation and less computational power than the APF. Although SMD can avoid some local minima, it does not guarantee that the global minimum is reached. In [BKMV04], Bray et al. incorporated SMD within a Particle Filter to form ‘smart particles’. After propagating the particles, SMD is performed and the resulting new particle set is included such that the original Bayesian distribution is not altered. As a particle method, it maintains multiple hypotheses needed to cope with clutter and occlusion, but reduces the number of particles needed. Figure 2.8 shows an example of 3D recovery using (note) structured light.
2.3 3D model-based tracking

Figure 2.8: A frame from a video sequence used by Bray et al. [BKMV04]. (a) and (b) show mesh created by the structured light, the red dots show the projected model with the tracking result obtained using: (a) APF and (b) Smart Particle Filter (SPF). (c) shows two views of the 3D model whose pose was obtained using SPF.

A deterministic alternative

While much effort has been made to explore stochastic approaches for human body tracking, Sminchisescu and Triggs have begun to explore such spaces deterministically, considered a way of avoiding entrapment in local suboptimal minima [ST02]. They address this problem by building ‘road maps’ of nearby minima linked by transition pathways – paths leading over low ‘passes’ in the cost surface, found by locating the transition state (saddle points with 1 negative eigenvalue) at the top of the pass and then sliding downhill to the next minimum. Their results have shown that their algorithm can stably and efficiently recover large numbers of transition states and minima, and also serve to underline the very large number of minima that exist in the problem of monocular 3D model based tracking.

2.3.7 Using data-driven dimensionality reduction

The use of articulated models simplify occlusion handling and allows the description of a larger number of hand poses. Although there has been an agreement that a 3D hand model should have at least 26 DOF, it is also clear that the configurations of muscles and tendons of the hand constrain the range of motion of each joint. For example, the fourth finger can not be flexed naturally without influencing the pose of the middle and small fingers because of interconnection of tendons (see Figure 1.2).

Heap and Hogg [HH96] represented hands as surface meshes extracted semi-automatically from 3D Magnetic Resonance Images. Since this is not based on an articulated model, the number of DOF of this representation is huge, but they have shown that by applying PCA to the point distribution data, the shape
deformation could be represented in a low dimensional space, as illustrated in Figure 2.9. For tracking, the outline of the hand mesh was projected into the image plane and edge measurements are used. The pose update was performed by solving a linear system of equations in least squares. Tracking could be achieved at 10 frames per second using a single camera, but ambiguous motions and self-occlusions were not successfully overcome.

![Image](image.png)

**Figure 2.9:** The first (a) and second (b) modes of variation of 3D hand point distribution model of Heap and Hogg [HH96]. (© [HH96], reproduced with permission.)

Wu et al. [WLH01], describe the space of possible hand configurations using a set of pre-defined states based on binary finger poses: fully flexed or stretched for each finger, giving a set of $2^5 = 32$ possible states of hand pose. Four of them were pruned because they were considered infeasible as most people cannot perform these hand poses naturally. Subjects were asked to move their hands to these 28 states while wearing a data-glove that acquires 15 DOF of the hand. Global position and orientation variations are not considered in this paper. The state space was reduced to 7 dimensions using PCA and it was demonstrated that the transitions between states follow linear paths in this space. Thus the hand pose is represented as a linear combination of the 28 states. An importance sampling approach is used for tracking. This shares some points with CONDENSATION, but the hypotheses are only generated along the nearest linear manifolds between two basis states, with some diffusion in the higher dimensional space. The 3D model is projected to the image as a cardboard model, and this method
combines edge measurements with a comparison between the area of the projected model and the hand silhouette image to compute the likelihood of hypotheses. The experiments show that, in comparison to standard CONDENSATION in the $\mathbb{R}^7$ space, this approach provides better results and longer latency requiring an order of magnitude less samples.

Using the same dimensionality reduction method and a similar image likelihood function, in Zhou and Huang propose an eigen-dynamics analysis method to learn the dynamics of natural hand motion as a high order stochastic linear dynamic system. This is used to build a dynamic Bayesian network to analyse the generative process of an image sequence of hand motion. In the inference phase, the hand motion is decomposed into global motion and finger articulation, and an iterative divide-and-conquer approach is used to track the hand. For global motion, the iterative closest point algorithm is applied, and for finger articulation, sequential Monte Carlo is used to sample in the manifold spanned by the learned dynamic model. This system was tested with synthetic and real data and accurate results were obtained even with partial occlusion and cluttered background, but the experiments do not show how the system performs when there are both global and articulated motion at the same time.

Other relevant work in this area is that of Kato et al. who reduces the state space dimensionality to 5D using ICA (independent components analysis), showing that it performs better than PCA, and Grochow et al. who represent the probability distribution function of the parameter space using a scaled Gaussian process latent variable model (SGPLVM) proposed in Law. All the parameters of the SGPLVM are learned automatically from the training data. They show that it is possible to optimise the PDF to describe new poses in real-time for applications of inverse kinematics systems. Although this method allows to represent the PDF at a low dimensional space through a non-linear projection, it does not restrict the configuration state. Poses that are very different from those in the training set can still be represented, but they have a very low PDF. The authors have used this method to represent styles of human movements and proposed a method to interpolate between styles (Figure 2.10).

### 2.3.8 Discussion

The key benefit of model-based trackers is that they permit, in principle, a comprehensive exploration of the space of possible poses – they really do describe the detail of all the degrees of freedom. However, the quality of the measurements to drive the model depends on the similarity between the model and the
2.4 Direct 2D view to 3D pose transformations

This section approaches view-based methods to estimate 3D pose. These are also known as discriminative methods, and provide a bridge between 2D methods and 3D model-based methods. They extract measurements from images which are linked to the kinematic chain representation of the object in 3D.
For instance, template matching or global image descriptors are used. Once the measurements are extracted, a pattern recognition method is applied and a 3D pose estimate is obtained as output. Since the measurements can be extracted without requiring a prediction of the state, discriminative methods can be applied from static images for pose estimation or to initialise 3D trackers.

As usual in pattern recognition, these methods require training. The training set is often an extensive collection of possible hand appearances associated with 3D poses that generate them. The most practical method to obtain a comprehensive training set is based on creating synthetic images using a hand model that is rendered at a range of 3D poses. So to restrict the training set to natural poses, data acquired from glove aided motion capture is used. In the case of whole body, publicly available datasets of human motions can be used.

An advantage of discriminative methods is that they do not require computation of projection and occlusion handling at the inference phase, as this is implicitly done in the generation of training samples. Another advantage is that since the inference uses the training set, these methods naturally incorporate data-driven motion constraints. This also allows to reduce the dimensionality of the parameters space. The obvious disadvantage is that the range of possible poses is limited by the training set and extrapolations are not usually successful. The same is true about camera views which are not included in the training set.

Two approaches encompasses the discriminative methods: classification-based and mapping-based, further described in Sections 2.4.1 and 2.4.2 respectively.

2.4.1 Classification-based methods

In the classification-based approach for 3D pose estimation, a large discrete set of 3D poses constitutes the set of classes. The image measurements are evaluated as an input to the classifier and a 3D pose is obtained as output. The ability to provide a 3D pose output is the key difference between these methods and the 2D appearance-based methods. Furthermore, the fact that the training set contains pairs of measurements and 3D poses is used to aid the search.

Usually only one sample image measurement is available for each class of 3D pose, thus variants of nearest-neighbour classifiers are commonly used. The massive number of classes make this a formidable classification problem, which is eased by avoiding exhaustive search. This can be done by the following
2.4 Direct 2D view to 3D pose transformations

methods: (i) performing coarse-to-fine search; (ii) grouping the training set by similarities in appearance and in 3D pose parameters; (iii) using motion priors and time sequence information. Sample research works of these methods are further detailed below.

**Coarse-to-fine search**

In the coarse-to-fine approach of Athitsos and Sclaroff [AS03], two similarity measures: the approximate directed chamfer distance and the line matching cost. A large pose database was used, containing over $10^5$ samples and the query could be made in 15 seconds, but the matching results were poor for real images: only 14% of the queries resulted in the best pose estimate, and even if all the 256 best matches are combined, the mean of correct matches among them is only 84%.

In [AASK04], an improvement was achieved by combining a large set of simple weak classifiers using BoostMap in the coarse search to select a subset of candidate matches. In the fine search, they used exhaustive chamfer matching. This reduced the query time to 2.3 seconds and improved that recognition rate to 95%. The quality of the classification results were judged by a human operator following the visual agreement between query and retrieval image.

**Grouping training samples for tree-based search**

One problem with exemplar-based matching is that the exemplar sets can grow exponentially with the number of degrees of freedom of the object. For this reason, Stenger et al. [STTC03] use a tree search (similar to Gavrila and Philomin’s method [GP99]) which leads to a dramatic reduction in the number of comparisons required for matching.

Another improvement is that [STTC03] also uses the probabilistic tracking framework proposed by Toyama and Blake [TB02], so the search tree works as a dynamic Bayesian network for motion estimation, as illustrated in Figure 2.11. But unlike Toyama and Blake, Stenger et al. perform the probabilistic tracking in the space of the kinematic parameters of the articulated object (joint angles, rotations and translations). An advantage of using a parametric model to generate templates is that less storage space is required, because a finer pose estimation can be obtained by generating new templates on line, as the leaf is reached. Furthermore, two poses that are distant in the parametric space can be close to each other in appearance [TSTC03]. For example, the appearance of the outline of a flat hand with the palm
facing the camera can be similar to that of the back of the hand facing the camera, but parametric-based clustering puts these two poses far apart.

![Tree-based estimation of posterior density](image)

Figure 2.11: Schematic example of tree-based estimation of the posterior density, obtained from [STTC03]. (a) Each node is associated with a non-overlapping set in the state space, defining a partition of the state space (here one DOF of rotation of the hand). The posterior for each node is evaluated using the centre of each set, and sub-trees with low posterior are not further evaluated. (b) Corresponding constant posterior density of the state $\theta$ given the measurement $D$, and piecewise constant approximation obtained.

To build the search tree, two methods have been evaluated Thananthan et al. in [TSTC03]. The first is based on the hierarchical $k$-means algorithm, which partitions the space as a multi-dimensional Voronoi diagram and the cluster centres are used as nodes in each level of the tree. In the second method, the dimensionality of the parameters space is reduced using PCA and the resulting space is partitioned by a regular hierarchical grid where, again, the centres of the obtained hyper-cubes are used as nodes in each tree level. Their experiments show that the tree obtained by both partitioning methods give qualitatively similar results and search time of around 2s per query. However the training process with the PCA-based method is much faster.

Based on the above search tree idea, in [STTC04] Stenger et al. proposed an alternative classification method which uses a multi-class cascade of classifiers for shape template matching. Unlike the normal use of boosting for single object detection, the cascade of classifiers is arranged in a tree order to recognise multiple object classes (hand configurations) hierarchically, as shown in Figure 2.12. Each weak classifier is trained to detect a single hand pose, if that pose is detected, the search continues for child classifiers that do a finer classification. As usual with boosting approaches, the main advantage is
its speed, but the number of classifiers needed grows exponentially with the dimensionality of the pose parameters, demanding much memory.

![Diagram of classifier cascade](image)

Figure 2.12: (a) Standard single class cascade of classifiers to detect an object – each classifier has a high detection rate and a moderate false positive rate. (b) Cascade of classifiers $C_i^j$ from [STTC04] for multiple classes $j$ in a tree structure (with levels indexed by $i$) – similar objects are grouped together and the classifiers on the leaves recognise single objects. A binary tree is shown here, but the branching factor can be larger than two. (©[STTC04], reproduced with permission.)

**Using spatio-temporal priors**

The tree-based system of Stenger et al. [STTC03] can also incorporate temporal priors. Given the partition of the state space, the state transition distributions $p(\theta_t|\theta_{t-1})$ are modelled as first order Markov processes, and the transition probabilities are computed by histogramming transitions in the training set. This allows the computation of the temporal priors $p(\theta_t)p(\theta_{t-1}|D_{0:t-1})$ (where $D$ are measurements and $\theta$ is the state vector) in a video sequence, which facilitates pruning the search tree, speeding up pose estimation as the motion follows a prediction. Although high accuracy can be obtained, the computational cost of this system is still too high for real-time applications. Using a relatively small range of hand poses, in [TSTC03] each frame takes 2 seconds to be processed in a 1 GHz Pentium 4.

An alternative is to use simpler image measurements and stronger temporal priors. In [FAK03], Fillbrandt et al. use a simple graph of transitions between states of the hand pose that restricts the search space, as only neighbouring states are checked. This graph was coded following transitions that happen in German sign language. A similar idea was implemented in [HSS02], where a simple moment-based
descriptor is used. In the learning stage, if the distance between the image descriptor of the current image and the previous state is greater than a threshold, a new state is built.

Brand [Bra99] uses ten scale-invariant central moments on low resolution silhouette images. A dynamical manifold is used for inference of trajectories. This is defined as a locus of all possible poses and velocity configurations, embedded in a higher-dimensional measurement space. The inference is a search for a sequence of events (path on the manifold) that best explains a sequence of observations. To model manifolds a method identifies neighbourhoods where the relationship of position to velocity is roughly linear. Each neighbourhood is described with a multivariate Gaussian PDF. The manifold is approximated by an HMM with each neighbourhood Gaussian being the output of a hidden state, and a transition topology specially matched to the dynamical structure of the manifold. The HMM is learnt using entropy minimisation which, unlike previous methods, leads to a model that does not get “lost” at crossings and gives a more compact and accurate representation. To handle rotations around the gravity axis, the HMM is replicated once for each view, re-estimating the output distribution of each view-specific HMM. The 3D pose results are, in most cases, qualitatively close to the actual pose of the input image sequence. But with evidence as weak as image moments, the learned prior dominates the reconstruction, so input images of poses that are not in the training set result in the nearest 3D pose in the training set, which, in many cases, is not accurate.

Hee-Deok et al. [YPL06] proposed a framework of HMM models for whole body gesture recognition which recognises continuously, without the need of gesture segmentation. This framework initially has an array of HMMs for meaningless actions followed by an array of HMMs for gestures that are recognised and the whole scheme is closed as a loop. This paper concentrates on gesture/action recognition, rather than low level vision, so it is based on accurate human motion capture data obtained from the system described in [HKL06]. The dimensionality of the pose parameters is reduced using Fisher discriminant analysis.

In order to get a more continuous (in terms of inter-class difference in the pose output) estimation of 3D poses from a discrete set of training appearances, Shimada et al. [SKS01] combine an appearance-based discriminative method with a three-dimensional generative tracker. In the first stage, the silhouette of the hand (segmented by threshold) is described using the normalised eccentricity, which is a position
and scale invariant descriptor. For rotation invariance, the maximal points are aligned with the training vectors for matching. Classification is sped up using an adjacency map and beam search, which is implemented in a distributed system. Once the appearance has been matched, its 3D pose combined is with the predicted pose in order to generate the next prediction using a motion model. The new prediction helps to speed up the appearance matching method by restricting the search area. This paper shows good qualitative matching results, which were obtained at video rate (30Hz) on a 6 node cluster, but it does not show results using the 3D motion prediction module.

2.4.2 Mapping-based methods

Mapping-based methods use pairs of image measurements and 3D poses to learn a continuous map between them. They can provide smooth pose estimation results rather than an estimate that is out of a discrete set. The results can be compared to an interpolation of the training data, but in some cases small extrapolations are also possible. Mappers can usually be implemented with parametric functions, which mean that their memory complexity is much lower than that of classification-based methods. In those cases, their evaluation does not require large numbers of comparisons, so their speed is also greater than that of classification-based methods.

Lin et al. [LWH01] modified the method of data-driven dimensionality reduction described in [WLH01] to create a mapping-based method. A feature vector built from measurements obtained from shape descriptors was acquired from each basis state in the training phase. In the application phase, this feature vector is acquired and its distance to each basic state is measured. This distance is taken as the weight of each state, determining a point in the state space, which is then lifted to the original 15 DOF configuration space to reconstruct the hand pose.

Shakhnarovich et al. [SVD03] introduced an algorithm that learns a set of hashing functions that efficiently indexes examples. The method uses local regression, which works as interpolated k-nearest neighbours and accounts for proximity not only in the 2D measurements, but also in the 3D pose parameters.

Prior to that, Rosales et al. [RASS01] proposed a system that uses a non-linear supervised learning framework, the specialised mappings architecture (SMA). As in Brand’s paper [Bra99], image moments are used as measurements: seven real-valued scale, translation and rotation invariant Hu moments
These are computed from hand silhouettes which are detected and tracked using a skin colour blob tracker that locates and refines the solution adaptively. A face detector is used to improve the initialisation of the skin colour detector. The pose estimation system consists of a set of 30 specialised forward mapping functions, each one built as a one hidden layer feed-forward network with 5 hidden neurons. These functions are learned using expectation-maximisation (EM). Each of them provides a mapping from the whole measurement space to the state space of 3D poses. To select the best solution, a feedback function takes the estimated pose, renders the 3D hand model and generate image measurements that are then compared with the input data. This method was evaluated quantitatively with a database of synthesised images generated using ASL gestures rendered at several orientations varying pan and elevation (the hand pose is described using 22 joint angles and two orientation parameters). This added up to 300,000 synthesised images, of which 8,000 were used for training and the rest for testing. The reported mean error was very small ($\approx 1^\circ$ to $3^\circ$), but the standard deviation was large enough to provide results that do not match the input ASL gesture entered. Qualitative results were also shown using real hand images.

In [MM01, MM02], Mori and Malik used shape context matching [BMP02] to locate the centre of limbs joints. The 3D pose is then estimated by using Howe et al.’s method [HLF99]. This is a Bayesian learning framework to recover 3D pose from known joint centres based on a training set of pose-centre pairs obtained from re-synthesised motion capture data.

A global image descriptor that is a simplification of shape contexts is used by Guan et al. in [HGT06], where the multi flash approach of [RTF+04] and [FTR+04] provides a clean depth discontinuity map, so the shape contexts describe a virtually noise free hand edges image. The mapping method used is based in self-organising maps.

In [AT04a], Agarwal and Triggs use a 100D global image descriptor based on a histogram of shape contexts of the silhouette contours. A human body model with 55 DOF is used to render training images and a regression-based method was used to learn the relation between image measurements $x$ and 3D poses $y$. Four regression methods were evaluated: (1) regularised least squares and (2) Relevance Vector Machine (RVM) [Tip01] regressors applied in both case to (a) linear and (b) Gaussian kernel bases. For synthetic images, resulting mean error in 3D pose estimation were: $(2a) > (1a) > (2b) > (1b)$, but
the difference between the best and the worst of them is only less than $3^\circ$. However, the implicit feature selection obtained by RVM regression gives much more sparsity, reducing the complexity of the pose estimation process: only 6% of the training examples were retained. They have shown good quantitative results on synthetic data: mean estimation error of $6^\circ$ over all joints for the Gaussian RVM (though many of the 55 DOF are inactive and it is not clear whether this is considered for this result). Only poor qualitative results were obtained for real images, and the demonstrative video shows reconstruction with many jitters along the sequence. Figure 2.13 illustrates a result of this tracker.

![Figure 2.13](image)

Figure 2.13: A sample result of [AT04b] (a) 3D human model used for training and its noise-free projected silhouette (b), which is used for training the regressor. (c) left: a test image obtained from [http://mocap.cs.cmu.edu](http://mocap.cs.cmu.edu); centre: background subtracted image segmentation result used for extraction of the shape contexts; right: reconstructed 3D pose. Note the difference between the subject and the training model and the amount of noise in the segmented image. (c) [AT04b], reproduced with permission.

This method has been modified to include a dynamical model with motion priors [AT04c] and has been embedded in a tracking framework combining dynamics from the previous state estimate with a special regressor to disambiguate the pose. Tracking is then formulated either as a single fully regressive model or by using the regression estimates in a multiple hypothesis tracker based in CONDENSATION [AT06b]. In contrast to Rosales et al. [RASS01], this method demonstrate an ability to deal with ambiguities in a probabilistic manner. A similar method was contemporaneously proposed by Sminchisescu et al. [SKLM05]. For hand tracking, Thayananthan et al. [TNS+06] extended the Tipping’s original Relevance Vector Machine method [Tip01] for multidimensional target spaces and multiple hypotheses. Unlike Agarwal and Triggs [AT05] regressor, this includes the hyper-parameters in the optimisation process.

In [AT06a], Agarwal and Triggs used SIFT features [Low04] computed on a regular grid on the
whole image. No segmentation is required, but the contribution of the background noise to the image descriptor is minimised by eliminating or downweighting background features using non-negative matrix factorisation [LS99], which is trained with features from clean foreground images. Pose estimation is then performed using the same unimodal regression method as in [AT04a], because the experiments only show estimation of the upper body pose, which is less ambiguous than the whole body. The experiments show that, for images with cluttered background, this method provides similar pose estimation performance to the method based on segmented silhouettes. The downside of this method is that it is not invariant to scale, rotation or translation, but it can be robust to some variation in clothing. In [AT06a], extensive experiments with synthetic images were performed, but only a few real image samples are shown. Both in the real and synthetic images shown, people wear tops with fairly uniform textures. The authors claim that better results can be achieved if larger training sets are employed.

2.5 A note on criteria for comparative evaluation of results

Despite a large body of work has been found in the literature, no standard methodology has been found to evaluate tracking and pose estimation results. For body tracking there are some human motion capture (HMC) data available publicly, (e.g. [Car]), but such data was used to generate synthetic images to which the tracker is applied, as the original HMC natural images are not available. An exception is the database described in [HKL06], which has human motion capture data with silhouettes and original images, but it is not publicly available. For hand tracking, there is no standard database or systematic evaluation method.

Ramanan and Forsyth [RF03] report tracking success whenever there is any overlap between a limb and the ground truth. This is probably very generous, but it is a good criteria for real-time applications on images with severe occlusions, fast movements and large acceleration. Most of the researchers have claimed that a qualitative visual agreement between the back projected models and the image is the most basic requirement of tracking performance. This is usually demonstrated with videos made available in the Internet.

The cost function that is used to minimise the state estimate of the trackers can provide a quantitative description of the tracking result. However, it does not provide a meaningful evaluation of the error of
the pose estimate.

*Track life* is the length of time that the tracker remains on target. Track loss occurs if the measured cost grows arbitrarily large because the model is no longer projected on the correct parts of the image. Track life can be used to validate the result, though it is not a strong criteria, because problems caused by singularities and deficiencies of the model may not be made explicit [Reh95].

For whole body tracking, Sigal et al. [SBR+04] were able to perform a quantitative evaluation using a professional marker-based motion capture system that was calibrated with the cameras used for tracking. This is the most accurate solution, but such professional systems are rarely available for research purposes mainly due to their cost.

Manually measured ground truth data has been employed by some authors (e.g. [BGP96]). Such measurements are usually obtained through mouse clicks in the position of the joints in the images and the use of a minimisation method to estimate the model posture from the measured positions. The obvious disadvantage of this method is that human operators are not reliable (or not available), specially for long sequences. The measurements can also be inaccurate because some joints may not be visible in all the images and determining the position of the joints is not always obvious.

Some researchers have used off-line processing using more computationally demanding parameters and multiple cameras to estimate the ground truth data. Such data is used to evaluate on-line real-time or monocular implementations (e.g. [FGTK02, TRMM01]). However, the reliability of such method is dubious if the same method is applied for on-line and off-line tracking.

A plausible alternative is to perform experiments in which the user is asked to touch known points in the world, as Bernardo et al. did in [BGP96]. But this does not accurately evaluate the estimation of all the joints of the hand.

Therefore, there is a demand for comparative evaluations of different methods for 3D hand tracking. While a standard framework or benchmark database is not defined, the evaluation method will be chosen according to the resources available and comparisons can be made between methods implemented within an institute, rather than globally.

This thesis shows comparisons of methods in Chapters 6 and 7. In the former, two generative methods are compared in terms of accuracy, efficiency and robustness. The comparison is based on their
formalism, quantitative tracking results on synthetic data, and time measurements. In the latter, a single view is compared with a multiple views discriminative method. That comparison is based on quantitative tracking results on synthetic data, qualitative results on real images and time measurements.

2.6 Summary and concluding remarks

This chapter reviewed the main approaches to hand tracking including a range of references from methods based on extracting meaning directly from low level image features to higher level methods. The main interest was on methods that estimate the hand pose in 3D in real-time. A lower level method to locate areas of interest is studied in Chapter 3 where it is identified that there has been a consensus that the most reliable cue for hand tracking is obtained by skin colour detection using classifiers applied to a colour space with brightness normalisation.

Two main approaches for 3D hand pose estimation have been identified: generative model-based and discriminative estimation methods. The use of a training set of natural hand poses is essential for discriminative methods. For model-based methods, it has been shown that such data is of great importance to reduce the dimensionality of the state space, reduce ambiguities and increase accuracy and reliability.

Temporal information and motion priors have originally been used only for model-based methods, but recent discriminative methods have shown that the use of such information reduces the estimation cost and the ambiguities.

Although discriminative (or “tracking as detection”) approaches are robust and have no latency limit, they do not make model-based methods obsolete. As shown later in Chapters 4, 6 of this thesis, model-based methods are view independent and less dependent on the training data. They can also provide a complete and continuous coverage to the parameters state. Model-based methods are easily scalable for multiple views and this has shown obvious improvements to the estimation results. But recently such approaches have been left aside for model-based methods. In discriminative methods they have only been explored modestly so far (e.g., [HSS02]). This thesis exploits multiple views for both generative and discriminative approaches. A novel multiple view discriminative method is described in Chapter 7 and comparisons with a single view implementation are shown.

Considering the literature, overall, some very good results have started to be shown with fast meth-
ods, but their robustness and accuracy have still not reached a point where a wide range of follow up applications could be successful. This shows that much improvement can still be achieved. Some contributions have been achieved in this thesis, which are described in the next chapters.