

Improving Sequential Monte Carlo Tracking by Bootstrapping

Thomas B. Moeslund
Laboratory of Computer Vision and Media Technology
Aalborg University
Niels Jernes Vej 14
DK-9220 Aalborg
Denmark
E-mail: tbm@cvmt.dk

Abstract

In model-based computer vision the problem of a high dimensional solution space often appears. The standard solution to this problem is to use a prediction followed by an iterative search or a Kalman filter. The drawback of both is the risk of ending up in a local extremum. Sequential Monte Carlo (SMC) methods have therefore been applied to estimate the entire posterior PDF. In this paper we suggest to apply bootstrapping to increase performance in SMC tracking. By bootstrapping we mean to track reliable low-level image features and use them to bootstrap the high-level model-based tracking. Concretely, bootstrapping allows a more compact state-space representation and a correction of the predictions and the process noise. The concept of bootstrapped SMC tracking is exemplified by monocular tracking of the 3D pose of a human arm with the position of the hand in the image as the bootstrapping information. In addition to bootstrapping, the paper also discusses different approaches for estimating the most likely state of the posterior PDF and suggests applying a method based on maximising a proximity likelihood function. Finally a number of tests are conducted and it is concluded that bootstrapped tracking is a promising approach for solving some of the inherent problems in model-based tracking in general and in SMC tracking in particular.

Keywords: Computer vision, human motion capture, maximum a posteriori.

1 Introduction

In the field of computer vision, the task of motion analysis is of great importance. Especially, methods such as structure from motion and tracking are considered important when solving general, as well as specific, computer vision problems. A special case of motion analysis is when the object to be monitored is known a priori, e.g., a vehicle or a human. The a priori knowledge allows for the usage of a geometric model of the object in the analysis process. This model can be used either to guide the motion analysis process, e.g., by constraining the diversity in the allowed motion patterns, or even more profoundly, by comparing the geometric model directly with the image data. The latter is known as model-based computer vision and operates by comparing different configurations of the model with the image data and let the current state of the monitored object be defined as the configuration most similar to the current image data.

Model-based tracking is a powerful approach for doing motion analysis. Furthermore, the approach assures that motion found in an image always corresponds to a legal configuration of the object. However, the state-space (or solution space) is often too high dimensional for an exhaustive search, i.e., too many different configurations need to be compared with the image data. A standard solution to this problem is to use a prediction to narrow down the solution. Either in the context of a state update, e.g., via the Kalman filter, or via an iterative algorithm. In case of the Kalman filter the current state of the tracked object is found as the weighted sum of the predicted state and the estimated state found from image data. The iterative algorithm starts with a predicted state and iteratively compares neighboring states from the solution space with the image measurement until an extremum is reached.

Both approaches are likely - sooner or later - to get stuck in a local extremum in the solution space, due to clutter in the background, wrong prediction, or when no image measurements of the tracked object can be found. Furthermore, when a monocular setup is applied to track a 3D object the image measurements are ambiguous by nature and only predicting one state will eventually cause the tracking to fail.

1.1 Multi-Modal Distributions

To handle the above mentioned problems a different strategy is needed. If the tracking problem is formulated in terms of Bayes' rule the problem can be defined as a matter of finding the maximum a posteriori (MAP), i.e., the most likely state given the measurements and the a priori knowledge. Assuming only one peak is present in the posterior probability density function (PDF) the problem has an analytic solution, the Kalman filter¹. However, in general this is not the case and multiple peaks should be expected. This, usually, leaves us with a very complex PDF of the posterior with no parametric representation. A solution to this problem can be found using a Sequential Monte Carlo (SMC) method. A Monte Carlo method represents the posterior PDF by a finite number of weighted state samples (known as particles) each selected from an *importance function* and weighted by the measurements. This sampling principle is known as *importance sampling*.

An SMC method is a Monte Carlo method operating on a time sequence of measurements [12]. Here the importance function can be defined from the posterior PDF predicted from the previous time instant. In other words, each of the most likely states in the posterior PDF in the previous time instant is sampled, predicted into the current time instant, and compared with the current image in order to obtain a weight. The weight reflects the similarity between the

¹Other assumptions are also introduced, e.g., a linear transition model and zero-mean Gaussian noise.

predicted state and the image measurements, i.e., the likelihood. The predicted states and their associated weights defines the estimate of the posterior PDF in the current time instant.

The SMC methods have been implemented in many different ways using many different names, see [12] for an overview. In computer vision SMC is perhaps best known as Condensation, see e.g., [2][3][10][13][19][27][29][32][33][38], due to the influence of the work by Isard and Blake [15][16][17][18]. But it can also be seen described as Multi-Hypothesis-Tracking [7][8] or Particle Filtering [6][11][21][30][31][37].

How effective the principle of SMC works depends on at least three issues: i) the number of particles used to estimate the posterior PDF, ii) the quality of the prediction, and iii) the MAP estimate.

The number of particles, N , can be tuned to a particular application or even changed during processing. In general N should be kept as low as possible since the computational demands of the algorithm growth exponentially with respect to N [9]. In fact N increases with both the dimensionality of the state-space and the covariance of the posterior PDF. In systems where the evaluation of the similarity between image measurements and a particular state is costly a low N is crucial to achieve a (close to) real-time algorithm. Different approaches have been suggested to limit N where the key principle is to make a local search in the state-space within a certain hypercube, see e.g., [7][8][37]. Each time a particle is predicted its predicted state is corrected to that state in the hypercube most similar to the image measurements. This forces particles to be more concentrated around peaks in the search space and thus fewer particles are required to estimate the posterior PDF. The approach works well reducing N but the downside is that true states (hypotheses) can be missed if they are within the same hypercube as another (potentially wrong) dominating hypothesis. So the trade-off is between small hypercubes with good accuracy but high N , and large hypercubes with less good accuracy but low N . A similar approach to reducing the required number of particles is the annealed particle filter [11] where the estimated posterior PDF is refined - meaning focusing the particles around the peaks - by re-sampling a number of times. Here the dilemma is not the size of the hypercube but how to choose the number of re-samplings and the diffusion of the particles in each re-sampling. In [33] another approach is suggested which seeks to avoid the problematic decision on the size of the hypercube. The approach is denoted Covariance Scaled Sampling and works by first finding the 'X' best local extrema in the solution space and then by including samples around each extremum in accordance with the uncertainty of each extrema defined by their respective covariances. This will focus the samples in the most likely regions and avoid the truncation of valid hypotheses near an extrema. The price is that additional samples are required.

The quality of the prediction is also an important issue in SMC. If the prediction is precise fewer particles are required to estimate an accurate posterior PDF. If, on the other hand, the prediction is less precise an artificially high process noise is required to allow the particles to diffuse in the state-space and estimate the posterior PDF [18]. This again requires a high N and results in poorer estimation of the posterior PDF. The obvious reason for not having a precise predictor is that it in general is difficult to model the dynamics in a particular tracking scenario, due to complex motions and lack of ground truth. An alternative approach to increasing the process noise is to let a certain amount of the particles be drawn from a uniform distribution and thereby be able to track non-predicted (non-modelled) events [10]. Of course this approach requires an increase in the number of particles. In [15] the notion of having two relatively simple dynamic models evaluated in parallel as opposed to one complex model is presented. Even though this might be a valid approach resulting in smaller N and precise predictions, the problem of lacking ground truth and the difficulty of too complex dynamics are still present and therefore most implementations of SMC still use simple dynamic models with artificially high

process noise.

The last issue in SMC is the quality of the MAP, that is, how the state of the tracked object is estimated at a particular time instant. Since the estimate of the posterior PDF is in the form of N weighted particles the representation is obviously non-parametric. The MAP is therefore estimated via moments, i.e., the mean and covariance of the posterior PDF. This works well if the posterior PDF is uni-modal, but as one of the key notions behind applying SMC in tracking is to allow multi-hypotheses, moments might not be the best choice! One solution is to make the posterior PDF parametric as in [7] where the posterior PDF is modelled by a sum of piece-wise Gaussians, those parameters are found from the predicted and weighted particles. Another approach is to smooth the entire sequence of posterior PDFs using both foresight and hindsight. This off-line approach has been implemented in [16] using some of the algorithms known from HMM.

1.2 The Content of this Paper

In this paper we focus on how bootstrapping can be applied to improve prediction in tracking algorithms in general and in SMC in particular. As the choice of N is not as critical in this work as in other systems we have not devoted any special attention to this issue. The issue of estimating the MAP will, however, receive some attention and an alternative approach will be suggested.

Concretely the paper is structured as follows. In section 2 we present a general framework for how bootstrapping can be applied in a tracking system. The rest of the paper will be devoted to exemplifying our approach in the context of tracking the 3D pose of a human arm by one camera utilising an SMC algorithm. Section 3 describes the geometric model of the arm utilised in this work, i.e., the state-space. In section 4 we describe the representation of the image data that will be used in a comparison with the predicted states of the state-space. In section 5 we describe how the bootstrapping is utilised in our context and introduce an alternative approach for estimating the MAP. Section 6 presents the results and section 7 discusses our findings.

2 Prediction and Bootstrapping

Predicting a state, $\vec{U}(t)$, from time $t - 1$ to time t can be done in many ways. However, it is usually done by adding a deterministic part, $\vec{D}(T)$, and a stochastic part, $\vec{S}(T)$, hence $\vec{U}(t) = \vec{D}(T) + \vec{S}(T)$, where T indicates dependencies of the entire past, hence $T = \{0, 1, \dots, t - 1\}$, and t indicates the present. $\vec{D}(T)$ consists of a motion model, $\vec{M}(T)$, which describes how the state evolves over time. $\vec{M}(T)$ contains a number of parameters whose current values are kept in $\vec{\omega}(T)$. $\vec{M}(T)$ is usually independent of time. $\vec{\omega}(T)$ is typically estimated in a recursive framework where it is assumed to be a first order Markov process, hence $\vec{\omega}(T) = \vec{\omega}(t - 1)$. In practise the deterministic part is normally defined as $\vec{D}(T) = \vec{D}(\vec{M}, \vec{\omega}(t - 1))$ [4].

A motion model that completely describes all aspects of how the state evolves over time can very seldom be set up. The stochastic part is therefore added. It models the errors in the motion model and is referred to as the process noise. $\vec{S}(T) = \vec{S}(\vec{N}(T), \vec{\phi}(T))$ where $\vec{N}(T)$ is the model of the process noise and $\vec{\phi}(T)$ is the current values of the parameters in this model. The process noise is often assumed to be independent of time and modelled as a Gaussian distribution. And as above a first order Markov process is assumed, hence $\vec{\phi}(T) = \vec{\phi}(t - 1)$. In practise the stochastic part is therefore normally defined as $\vec{S}(T) = \vec{S}(\vec{N}, \vec{\phi}(t - 1))$, where \vec{N} is a multivariate Gaussian distribution.

The motion model and the different parameters can be learned through training, see e.g., [4]. Learning the parameters through training can be difficult due to lack of ground-truth data. An alternative is therefore to estimate $\vec{\omega}(t-1)$ and $\vec{\phi}(t-1)$ for each image. However, this can also be very problematic. They require complete knowledge of the state at $t-1$ and as multiple hypotheses do occur, finding *the* correct state for each image is not always possible. The standard way of finding the correct state is by estimating the maximum a posteriori (MAP) and this is not trivial in the context of an SMC algorithm, see section 5.2.1.

2.1 Bootstrapping

When tracking an object it is sometimes possible to recognise parts of the object prior to tracking. For example, in the context of tracking the 3D human figure in a monocular image sequence it is in general difficult to find robust features to track, however, some features can actually be tracked independent of others. These are: the face/head, the hands, the feet, and in some cases also other distinct points, e.g., arm pits, shoulders, and crotch.

Say we are able to find one of these features, denoted by $\vec{\beta}(t)$. This would allow a comparison between $\vec{\beta}(t)$ and $\vec{U}(t)$ resulting in an estimate of (parts of) the prediction error, i.e., $\vec{\phi}(t)$. Applying $\vec{\phi}(t)$ as opposed to $\vec{\phi}(t-1)$ obviously gives a far better estimate of the stochastic part. We denote the new estimate with a plus, $\vec{S}(T)^+ = \vec{S}(\vec{N}, \vec{\phi}(t))$.

Furthermore, $\vec{\beta}(t)$ also contains information that can be used to bias the deterministic prediction, or more precisely $\vec{\beta}(t)$ can correct (parts of) the predicted deterministic part. That is, given $\vec{\beta}(t)$ we can estimate $\vec{\omega}(t)$ and apply this instead of $\vec{\omega}(t-1)$. We denote the new estimate of the deterministic part as $\vec{D}(T)^+ = \vec{D}(\vec{M}, \vec{\omega}(t))$.

So instead of predictions based on estimates at time $t-1$ we now use our estimates from time t , $\vec{\beta}(t)$, to correct our predictions, altogether providing a better result.

We denote this approach *bootstrapped tracking*. The success of this approach depends on how much information is carried in $\vec{\beta}(t)$, i.e., how many of the state's parameters can be corrected, and how much this information can prune the solution space.

As bootstrapped tracking delivers precise predictions (corrections) simpler motion models can be applied. The often difficult and tiresome task of acquiring training data and learning the motion model and its parameters can thus be avoided altogether.

3 Modelling the Arm - The State-Space Representation

A concrete tracking problem is required in order to evaluate the notion of bootstrapped tracking. We use the problem of monocular tracking of the 3D pose of a human arm as a case study.

In this work we find the position of the hand in the images, $[h_x, h_y]^T$, using colour segmentation [26] and let this be our bootstrapping information, hence $\vec{\beta}(t) = [h_x, h_y]^T$. In the context of $\vec{\beta}(t)$ a geometric model of the arm needs to be defined. Before doing so we introduce the assumption that the hand is a part of the lower arm and that the 3D position of the shoulder is known, e.g., via the algorithm described in [26].

The human arm is usually modelled as either the 3D positions of the elbow and hand, or by four angles together with the length of the upper arm (A_u) and the lower arm (A_l). Both representations require six parameters. If we however assume the length of the two arm segments to be known, we only need four angular parameters. These can be, e.g., angles around fixed axes or Euler's angles [25]. The latter is often applied as it is similar to the anatomic joint angles

$$\vec{H}(d, \vec{\beta}(t), \vec{\Phi}) = \vec{P}(\vec{\Phi}) + d \cdot \vec{F}(\vec{\beta}(t), \vec{\Phi}) \quad (1)$$

where $\vec{P}(\vec{\Phi})$ is the optical centre of the camera, and $\vec{F}(\vec{\beta}(t), \vec{\Phi})$ is the unit direction vector of the line. Equation 1 only contains one free parameter, namely d . In this work, however, H_z is used as the free parameters since it has a more intuitive interpretation. So for each value of H_z d and therefore also H_x and H_y are uniquely determined. By applying equation 1 to the screw axis representation we can eliminate the parameters H_x and H_y which leave us with just two parameters, namely α and H_z to model the configuration of the arm. We denote this novel model the *local screw axis model*. For each new image a unique instance of the solution space exists, hence the name "local". In figure 2.A the solution space is illustrated in the 3D Cartesian coordinate system for one image where the hand is located somewhere on the line, l . That is, the shape represents all possible positions of the elbow, given that the hand is located somewhere on the line and the shoulder is located in the position indicated by a '*'. In figure 2.B (ignoring the dotted lines) the local solution space is illustrated for the parameters of the local screw axis model. Clearly this model has a very simple and compact representation of the solution space compared to the complex shape of the solution space in figure 2.A².

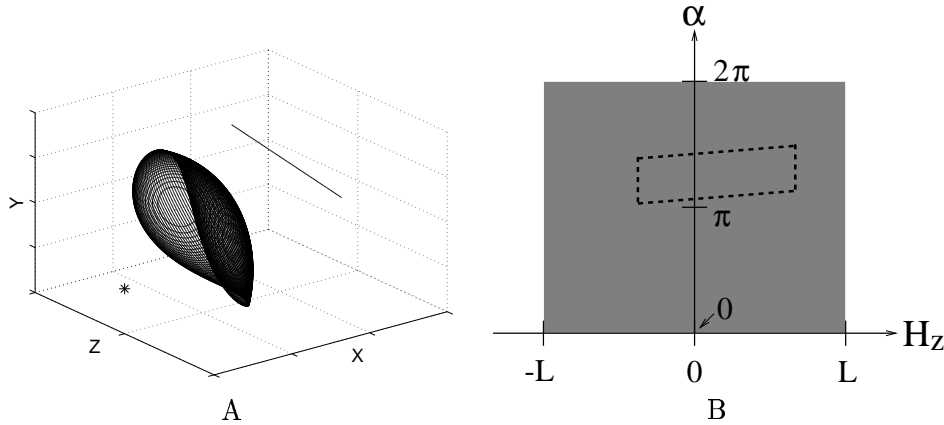


Figure 2: A: The solution space for one image in a 3D space. The line indicates the possible positions of the hand. The "surface" illustrates the solution space for 80 discrete position of the hand on the line, hence 80 circles. The '*' is the position of the shoulder, i.e., (0, 0, 0). B: The solution space in the local screw axis model, where $L = A_u + A_l$. The region within the dotted lines illustrates a typical solution space after pruning.

In this compact representation α is bounded by one circle-sweep ($0^\circ - 360^\circ$) while H_z is bounded by \pm the total length of the arm. Given a total arm length of say $60cm$ and a resolution of 1° for α and $1cm$ for H_z we have a solution space containing $4.32 \cdot 10^4$ different solutions. A large reduction in the size compared to that produced by Euler's angles, but still a large space. We therefore introduce kinematic constraints to prune the solution space. For example, the arm can not bend backwards at the elbow. These constraints result in a minimum pruning effect of 75% and an average pruning effect of 91.9% [22]. In figure 2.B the average number of non-pruned configurations is illustrated as the region within the dotted lines.

²Not to mention that of the four Euler's angles!

4 Image- and Object Representations

As the context of this work is model-based tracking we need a way of comparing the image data and the model data. A large part of this problem is to find a suitable representation of both the model and image data. The two representations need to be similar in order to do the comparison. Typical image representations are 2D anatomic points (hands, head, feet, elbow, etc.), edges, silhouettes or 2D skeleton. Typical object representations are 3D anatomic points, 3D skeletons, 2D patches, and volumetric body parts [24].

In general it can be stated that the higher level the representation of the image has the simpler the object representation needs to be. For example, in [1] the images are processed until the skeleton of the human figure is extracted, that is, a high-level representation allowing a simple skeleton representation of the object model. In [20] the silhouette of the human figure is used as low-level image representation. To compare an object model data with the silhouette a high-level volumetric model of the object is required.

In this work we use a high-level image representation and a low-level object representation. We represent the image data by the orientations of the upper and lower arm in the image, i.e., a high-level representation. The local screw axis model needs not be enhanced as it can be mapped directly into orientations in the image given the calibration parameters, i.e., a low-level representation.

4.1 Estimating the Orientations in the Image

We estimate the orientations of the upper arm, θ_u , and lower arm, θ_l , respectively, based on edge pixels. As our input images contain background clutter and non-trivial clothes we utilise temporal edge pixels. That is, we find the edge pixels in the current image using a standard edge detector and AND this result with the difference image achieved by subtracting the current- and the previous image. Figure 3.A shows a typical input image. In figure 3.B the temporal edge pixels for this image are shown. Those pixels actually belonging to the arm will be located in four classes, two for the upper arm and two for the lower arm, respectively. Our system does not impose restrictions on the clothes of the user. The clothes will in general follow gravity, hence the two classes of pixels originating from the upper sides (with respect to gravity) of the upper- and lower arm will model the structure of the arm better, see figure 3.B. We therefore only consider temporal edge pixels located on the "upper" sides. Concretely we define "upper" and "lower" via two lines described by the position of the shoulder and hand in the image, together with a predicted position of the elbow.

As we wish to estimate θ_u and θ_l independently we separate the temporal edge pixels into two groups, one for the upper arm and one for the lower arm. This is done by calculating the perpendicular distance from each pixel to the two lines. As the prediction of the position of the elbow is uncertain we ignore all pixels within a certain distance from the predicted position of the elbow. Furthermore we ignore all pixels too far away from both lines. When no predictions are available different possible positions of the predicted elbow are investigated until two representative groups are obtained.

Estimating the orientation of a straight line from data can be carried out in different ways, e.g., via principal component analysis or linear regression. However, as we will not model the distribution of the orientations via Gaussians we can not apply these methods. Instead we apply a dynamic variant of the Hough transform - the dynamic Hough transform (DHT). It estimates the likelihood of each possible orientation, hence allowing multiple peaks in the observation PDF. The choice of the DHT is furthermore motivated by the fact that it adapts to the data.

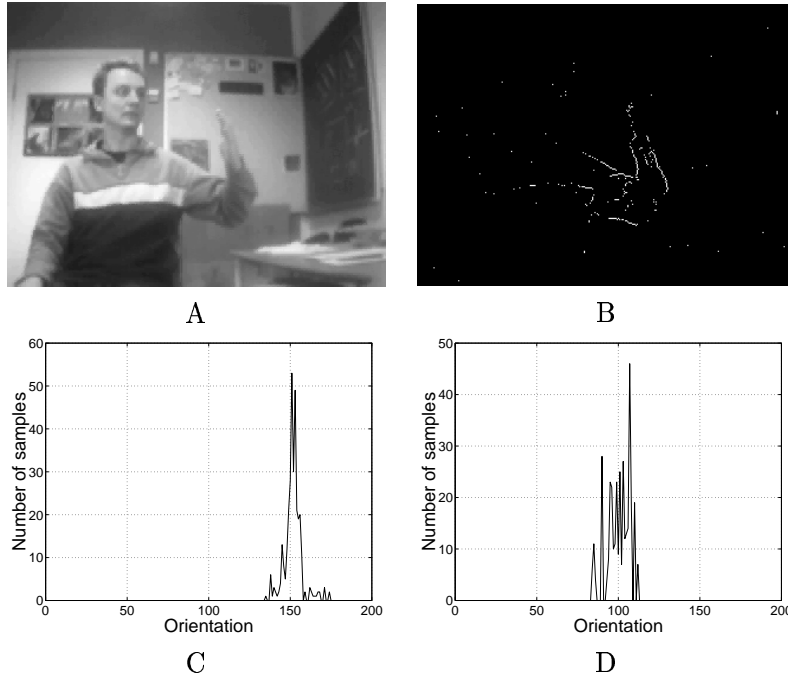


Figure 3: A: A typical input image (shown in B/W). B: The temporal edge pixels. C: The PDF of the orientation of the upper arm. D: The PDF of the orientation of the lower arm.

The DHT randomly samples two pixels from one group and calculates the orientation of the line spanned by the two pixels. The more times the groups are sampled the better the estimation of the PDF. On the other hand many samplings also lead to large processing time as is the case for the standard Hough Transform. The sampling is therefore terminated as soon as the variance of the PDF is stable. To evaluate the stability of the variance after n samplings the variance of the last j variances is calculated as

$$\nu_{jn}^2 = \frac{1}{j} \sum_{i=n-j}^n (\sigma_i^2 - \mu_{jn})^2 \quad (2)$$

where σ_i^2 is the variance after i samplings and μ_{jn} is the mean of the last j variances.

The stop criterion is defined as the number of samplings, n , where the last j samplings are within the interval $[\mu_{jn} - \lambda, \mu_{jn} + \lambda]$. The distribution of the last j variances will in general follow a Uniform distribution. The theoretical variance of such a distribution in the given interval can be estimated as $\lambda^2/12$ [28]. When the mean of the variances, μ_{jn} is large it indicates large uncertainty in the PDF, which again indicates weak lines in the temporal edge image. A stable variance for such a PDF tends to required a larger value of λ compared to an image with stronger lines. To account for this difference λ is defined with respect to μ_{jn} as

$$\lambda = \frac{\mu_{jn}}{\gamma} \quad (3)$$

where γ is found empirically. Setting the estimated variance equal to the theoretical variance yields $\lambda = \nu_{jn} \sqrt{12}$. Inserting this result into equation 3 and writing it as an inequality yields

$$\nu_{jn}^2 \leq \frac{\mu_{jn}^2}{12 \cdot \gamma^2} \quad (4)$$

Altogether the stop criterion is found as the smallest n for which inequality 4 is true. To speed up the calculations the variance is not recalculated after each new sampling, but rather for every 10th sampling.

Using the above described procedure we obtain two independent PDFs, one for the upper arm, $P_u(\theta_u)$, and one for the lower arm, $P_l(\theta_l)$. Examples of these are illustrated in the figures 3.C and 3.D. Different number of samplings might have been used to estimate the two PDFs. The accumulated probability mass for each PDF is therefore normalised to 1. In terms of the SMC algorithm the two normalised PDFs are the weighting functions, used to estimate the observation PDF, see below.

5 Bootstrapping the SMC Algorithm

The SMC algorithm is defined in terms of Bayes' rule and using the first order Markov assumption. That is, the posterior PDF is equal to the observation PDF multiplied by the prior PDF, where the prior PDF is the predicted posterior PDF from time $t - 1$:

$$p(\vec{X}_t | \vec{\theta}_t) = p(\vec{\theta}_t | \vec{X}_t) p(\vec{X}_t | \vec{\theta}_{t-1}) \quad (5)$$

where \vec{X} is the state, hence $\vec{X} = [\alpha, H_z]^T$ and $\vec{\theta}$ is the image measurements, hence $\vec{\theta} = [\theta_u, \theta_l]^T$. The predicted posterior PDF is defined as

$$p(\vec{X}_t | \vec{\theta}_{t-1}) = \int p(\vec{X}_t | \vec{X}_{t-1}) p(\vec{X}_{t-1} | \vec{\theta}_{t-1}) d\vec{X}_{t-1} \quad (6)$$

where $p(\vec{X}_t | \vec{X}_{t-1})$ is the motion model governing the dynamics of the tracking process, i.e., the prediction, and $p(\vec{X}_{t-1} | \vec{\theta}_{t-1})$ is the posterior PDF from the previous frame. As described in 1.1 the SMC algorithm estimates $p(\vec{X}_t | \vec{\theta}_t)$ by selecting a number, N_i , of (hopefully) representative states (particles) from $p(\vec{X}_{t-1} | \vec{\theta}_{t-1})$, predicting these using $p(\vec{X}_t | \vec{X}_{t-1})$, and finally giving each particle a weight in accordance with the measurements, i.e., $p(\vec{\theta}_t | \vec{X}_t)$. So, as explained earlier, a key issue is to have a precise prediction. Referring back to section 3 we have $\vec{\beta}(t)$ equal to the position of the hand in the image. To apply bootstrapping we need to define $\vec{D}(T)^+$ and $\vec{S}(T)^+$.

At this point it might be in order to emphasise that our state-space model is in the two parameters α and H_z , but all calculations are done in anatomic parameters. That is, the 3D position of the elbow, \vec{E} , and the 3D position of the hand, \vec{H} . These two representations co-exists and their relationship is illustrated in the figures 2.A and 2.B. The entities $\vec{D}(T)^+$ and $\vec{S}(T)^+$ are defined in terms of the anatomic parameters. We will do this first for the position of the hand, \vec{H} , and then for the position of the elbow, \vec{E} .

The correction of the prediction of \vec{H} is based on the notion of combining the predictions and the image measurements. In figure 4 the predictions are illustrated using subscript 'p' while the corrected predictions are illustrated using subscript 'c'.

Since we know the camera ray through the hand in the current image, l , we can correct the prediction by projecting the predicted position of the hand, \vec{H}_p , to the line, l . The projected prediction is denoted \vec{H}_1 and calculated as $\vec{H}_1 = \vec{P}(\vec{\Phi}) + ((\vec{H}_p - \vec{P}) \cdot \vec{F}(\vec{\beta}(t), \vec{\Phi})) \vec{F}(\vec{\beta}(t), \vec{\Phi}))$ where $\vec{P}(\vec{\Phi})$ and $\vec{F}(\vec{\beta}(t), \vec{\Phi})$ are the line parameters defined in equation 1. The stochastic prediction models the process noise by diffusing the deterministic prediction. As we know the hand is on the line, l , we diffuse it by randomly sampling from a Gaussian distribution located

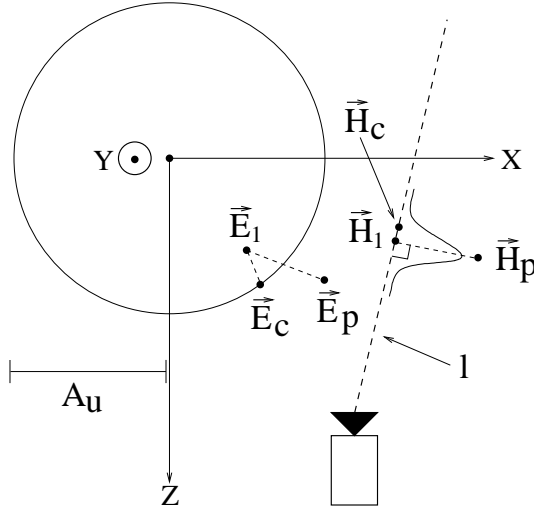


Figure 4: The shoulder coordinate system seen from above. The circle illustrates the sphere that defines the possible positions of the elbow. The large dashed line indicates a camera ray through the hand. See text for a definition of the parameters.

along the line, l , see figure 4. The mean of the Gaussian is defined by \vec{H}_1 and the standard deviation controlled by the error vector, hence the standard deviation = $k_1 \cdot \|\vec{H}_p - \vec{H}_1\|$, where k_1 is a predefined constant. After this operation we have the corrected prediction of the hand, \vec{H}_c . The difference between the predicted and corrected vectors yields a measure of the prediction error (innovation), denoted \vec{H}_e and calculated as $\vec{H}_e = \vec{H}_c - \vec{H}_p$.

The predicted position of the elbow can not directly be bootstrapped by $\vec{\beta}(t)$. However, we know it is likely to have a predicted error closely related to that of the hand as the hand and elbow are part of the same open-looped kinematic chain. We therefore calculate the corrected position, \vec{E}_c , by first adding the predicted error of the hand to the predicted value of the elbow, yielding $\vec{E}_1 = \vec{E}_p + \vec{H}_e$, and then finding the point closest to \vec{E}_1 that results in a legal configuration of the arm. In mathematical terms $\vec{E}_c = \arg \min_{\vec{E}} \|\vec{E} - \vec{E}_1\|$ subjected to the constraints $\|\vec{E}\| = A_u$ and $\|\vec{E}\vec{H}_c\| = A_l$. The solution to this problem can be found in [23].

As the error vector has already been subjected to diffusion we do not introduce yet another diffusion of the position of the elbow.

Evidently the prior PDF in the SMC algorithm will be much more accurate when applying the bootstrapping compared to the standard approach. And this is true even with a very simple motion model. This observation results in two things. Firstly, we use a first order linear motion model for both the hand and the elbow, hence $\vec{x}(t) = \vec{x}(t-1) + \vec{d}(t-1)$, where the latter term is the displacement in the previous image. This circumvents the often complicated task of acquiring sufficient ground truth training data and learning the model dynamics. Secondly, we conduct prediction at particle level, i.e., different motion parameters (displacements) for each particle. Obviously this means that the parameters of the motion model do not need to be learned, hence no training. Also, this procedure allows better predictions as the parameters of the motion model are local, hence avoiding the relationship between the estimated MAP and the prediction. This is desirable because the MAP is, in general, hard to estimate (more details below) and multiple hypotheses have different motion models.

5.1 Summary of Algorithm

To clarify our approach the different steps in the algorithm are described below. Note that the posterior PDF is represented as an unordered list of particles, each containing the following information: $\vec{a}_i = [\pi_i, c_i, \alpha_i, \vec{H}_i, \vec{V}_{Hi}, \vec{E}_i, \vec{V}_{Ei}]$, where π is the weight, c is the accumulated weight, α is the first parameter in the local screw axis model, \vec{H} is the 3D position of the hand, \vec{V}_H is the displacement of the hand, \vec{E} is the 3D position of the elbow, and \vec{V}_E is the displacement of the elbow. For each particle the following is done:

1. Sampling

One particle is sampled from the posterior PDF at time $t - 1$. The sampling is carried out according to the probability of each sample, i.e., its weight, π_i , and accumulated weight, c_i , see [17] for further details.

2. Prediction

The position of \vec{E}_i and \vec{H}_i are predicted using a motion model with local parameters, i.e., \vec{V}_{Hi} and \vec{V}_{Ei} .

3. Bootstrapping

The predicted values of the hand and elbow are corrected according to $\vec{\beta}(t)$ as explained above. The predicted values are mapped into the local screw axis model (α, H_z) and checked against the pruned part of the solution space, see figure 2.B. The local parameters of the motion model are calculated, i.e., \vec{V}_{Ei} and \vec{V}_{Hi} are updated.

4. Weighting

The corrected positions of the hand and elbow are mapped to two orientations in the image, one for the upper arm, θ_u , and one for the lower arm, θ_l . The weight of this particular particle is now calculated as $\pi_i = P_u(\theta_u) + P_l(\theta_l)$.

After the above steps have been conducted N times the weights are normalised, $\sum \pi_i = 1$, and the cumulative probabilities, c_i , are calculated as $c_0 = 0$, $c_i = c_{i-1} + \pi_i$.

Note that step three is the way the bootstrapping approach enhances the standard SMC algorithm.

5.2 Issues in the SMC Algorithm

Three issues always appear when implementing an SMC algorithm. Firstly, the value of N needs to be decided, secondly a procedure for initialising the SMC algorithm is required, and thirdly a procedure for estimating the MAP is required.

As already mentioned N is not as critical in this work as in other applications utilising an SMC method. The main reasons are the low dimensionality of our state-space and the relatively low complexity involved in comparing a particular state with the image measurements. We have therefore not devoted any special attention to the problem of selecting N . The choice of N is application dependent or can be found in a training phase as was done in this work. Methods for lowering N in the context of our approach will be discussed in section 7.

By initialising an SMC algorithm we mean to define the prior PDF in the first image where no predictions are available. We delay the algorithm one image and find the posterior PDF in the first image using our bootstrapping information. We use $\beta(0)$ to prune (α, H_z) and weight each non-pruned parameter set equally. This procedure forces the algorithm to converge much faster than given equal weights to all possible instances in the solution space and especially if the solution space is defined by the Euler's angles or any other four-parameter representation.

5.2.1 Estimating the MAP

Finding *the* correct state in a particular time instant is an important problem when an explicit representation of the tracked object is required. That is, when at some point all those fancy state-space tracking algorithms developed in different research-labs around the world are to be applied in real-life applications we have to be able to estimate the actual pose at a particular time instant.

The most common method to estimating the current state is by finding the weighted average of all sampled particles, i.e., the first moment [17]. This is simple, but has the major drawback that it assumes the posterior PDF to be uni-modal. In situations where this assumption do not hold, the estimated state might be located in a region where no samples are present at all or, even worse, in an impossible region of the solution space, i.e., a pruned region. In figure 5 an example of this is shown. The figure shows a (designed) 1D posterior PDF represented by weighted particles. The state indicated by ' \mathbf{x} ' is the first moment. Clearly this is an incorrect solution in this case. A slightly better solution in this line of thinking is to define the current state as the particle closest to the first moment.

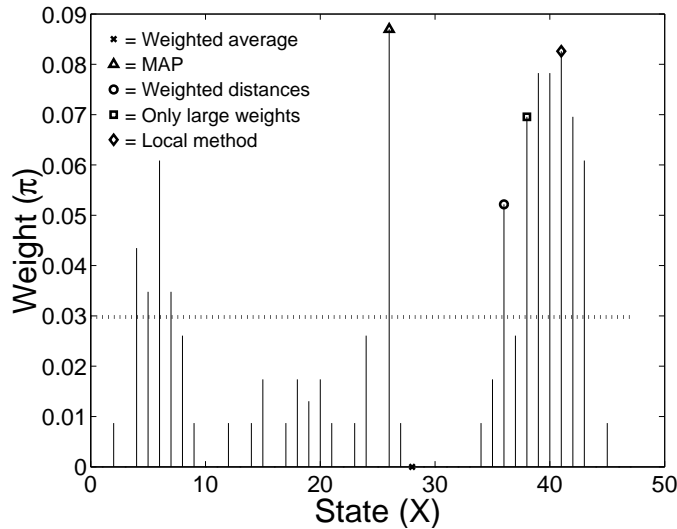


Figure 5: A posterior PDF represented by particles. The symbols refer to different estimates of the current state.

Another common approach is to estimate the current state as the particle with the highest weight. As the weight of a particular state might be the sum of several particles this is a very sensitive approach. A better solution in this line of thinking is to estimate the current state as the state with the highest weight, indicated by ' Δ '. This is exactly the MAP defined in section 1.1.

The MAP is a statistically sound concept that is well-suited in the context of Bayes' rule. However, in the context of estimating the current pose, i.e., the most likely configuration of an object, the MAP might not be the best choice. First of all the posteriori PDF is discrete. If the PDF was a continuous function the MAP would always be the state with the highest probability. However, in all practical matters the PDF is discrete and the MAP can therefore change due to different resolutions, i.e., the MAP is dependent on the resolution. Especially, in the case of multiple modes the MAP can change significantly.

Secondly, the state with the highest probability might not always represent the "best" state.

In figure 5 the MAP is represented as ' Δ '. Clearly a better estimate of the current state can be found if also the probability in the proximity is considered. In the figure a more dense probability mass is present to the right in the figure suggesting that a better estimate might be found here compared to the MAP.

Thirdly, the problem of an explicit representation of multiple-hypotheses. Clearly, SMC can represent multiple-hypotheses in an implicit manner, but how would we answer the following question. Which are the five most likely configurations of the objects in a particular instant of time? Applying the MAP we could find the five states with the highest probabilities. However, these are likely to originate from just one or two modes in the posterior PDF, i.e., one or two configurations, and by the question was meant, the five most likely *and* different configurations, i.e., different modes.

For the three above mentioned reasons we seek an alternative to the MAP that takes the probabilities in the proximity into account. An estimate of the current state which takes the probabilities (weights) of other states into account can be defined as the particle the state of which minimises the sum of the weighted distances between this state and all others states (particles).

$$\text{state} = \arg \min_{\vec{X}} \sum_{j=1}^N \pi_j \left\| \vec{X} - \vec{X}_j \right\| \quad (7)$$

This expression finds the optimal solution in terms of the weighted sum of absolute differences and is illustrated as state ' \circ ' in figure 5. In general, expression 7 gives a better estimate of the current state compared to the above mentioned methods. However, it has two major drawbacks; it is computationally demanding and tends to favour particles close to the first moment.

The high complexity is due to the high number (N^2) of weighted distances that needs to be calculated. To avoid a computational explosion only those particles having a large weight, suggesting that they are part of a large peak, are investigated. That is, particle a_i is investigated if $\pi_i > \frac{k_2}{N}$, where the constant k_2 directly determines the number of particles to be investigated, i.e., the computational complexity. In practise a new list containing all particles with a high weight is constructed and used instead of the original list containing all particles. In the example in figure 5 all particles above the dotted line are considered. The state found in this manner is illustrated in figure 5 as ' \square '. Besides lowering the complexity this method actually also "filters" the posterior PDF by removing all particles with a small weight and hereby improving the result, see figure 5.

Even though only particles having a large weight are considered, expression 7 still tends to favour particles close to the first moment. The reason being that expression 7 is a global method. To avoid this problem we instead suggest the local method in expression 8, which only considers the particles close to the particle being evaluated. Note that only those particles having a high weight are considered. Expression 8 defines the current state to be the state where the probability density in the proximity is highest. The state found using this expression is illustrated by ' \diamond ' in figure 5. We denote this approach MOLAP, most likely a posteriori.

$$\text{MOLAP} = \arg \max_{\vec{X}} \sum_V \pi_j \quad (8)$$

where $V = \left\{ j \in [1, N] \mid \text{abs}(X_k - X_{k,j}) < \Omega, \forall k \right\}$, and $k \in \left\{ 1, 2, \dots, \text{dim}(\vec{X}) \right\}$ is an index into the state vector, and Ω defines the proximity threshold.

In general expression 8 is an accurate estimation of the current state and since it considers the proximity it handles the first two problems mentioned above. The third problem is handled by ignoring all particles contributing to the best MOLAP, when searching for the second best MOLAP, etc. Furthermore, the expression also ensures that the estimated state is always in a non-pruned region of the state-space. Finally, the expression allows for a parametric representation of the posterior PDF. We could represent each mode in the posterior PDF by a Gaussian with mean given by the MOLAP and covariance given by the sum in expression 8. The 'X' best MOLAPs (modes) could then be found and the posterior PDF represented by the sum of the Gaussians representing the best modes.

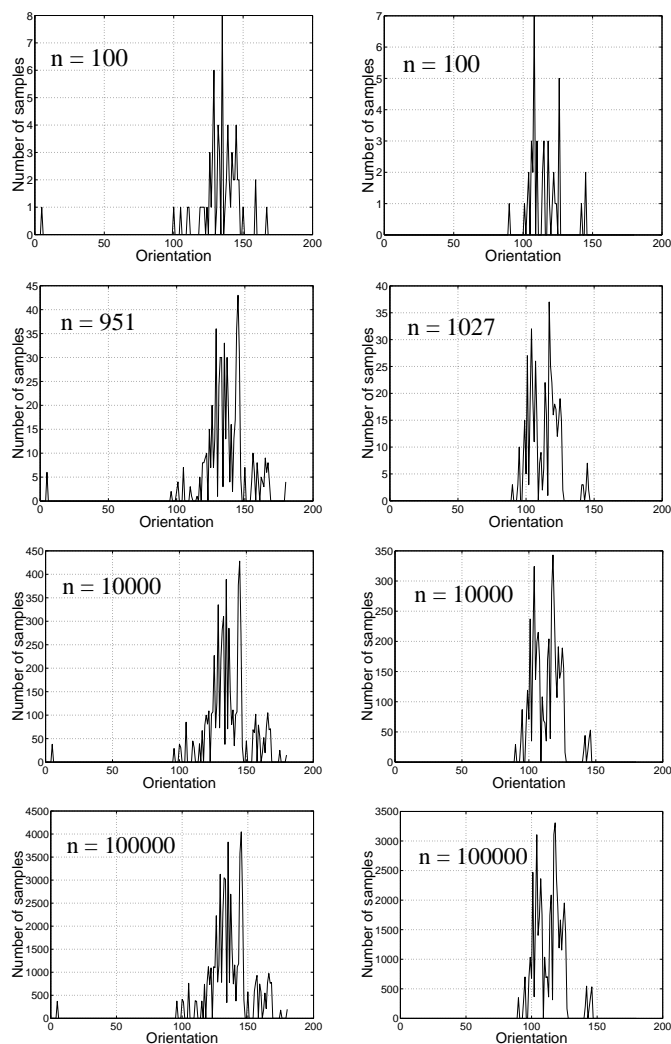


Figure 6: The estimated PDFs after n samples for a particular image. The left column is for the upper arm and the right column is for the lower arm.

6 Results

In this section we will show some results related to the different topics described in this paper. These are the dynamic stop criterion for the DHT, the choice of N , the procedure for estimating

the MAP, and finally the results of tracking with- and without bootstrapping are shown. Note that the results are not commented and discussed in this section, but rather in the next section.

In figure 6 the PDFs of the orientations of the upper- and lower arm are shown for the same image but for different number of samples in the DHT, i.e., different n . In figure 7 the variances of the PDFs for $n \in [0, 10000]$ are shown. The choices of j and γ (see section 4.1 for definitions) directly control the computational complexity of the DHT. In this work we have set $\gamma = 0.05$ and $j = 100$, resulting in $n \in [500, 1500]$. Obviously these values need to be tuned to the particular application at hand.

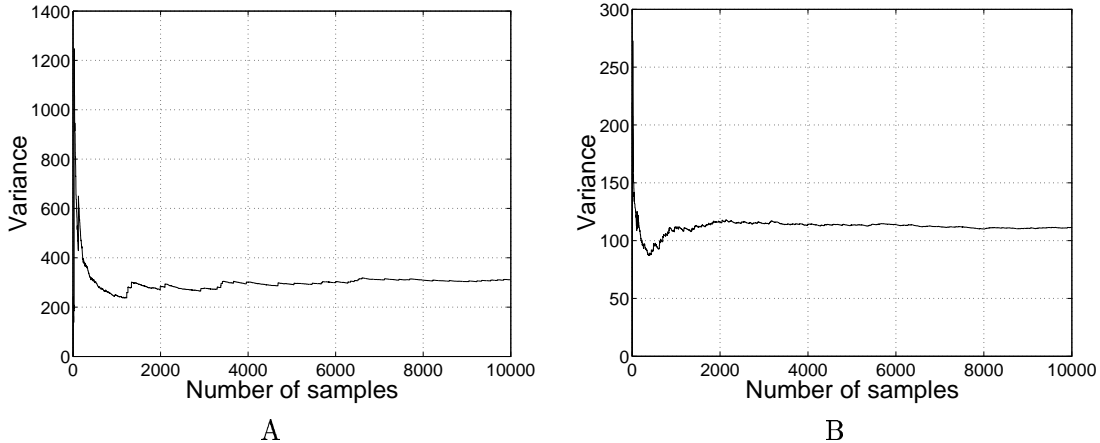


Figure 7: The variance for the PDF of the upper arm (A) and lower arm (B), respectively, as a function of the number of samples.

In figure 8 the posterior PDF for the image in figure 3.A is shown for three different values of N . The bottom figure represents the "true" posterior PDF and is estimated by setting $N = 10000$. The two other figures are generated from $N = 10$ and $N = 100$, respectively. For all calculations of the posterior PDF $k_1 = \frac{1}{2}$. See section 5 for a definition of k_1 .

The next test is conducted to investigate the proposed way of estimating the current state, i.e., the MOLAP. In figure 8 the current states are estimated in the standard way using a weighted average, illustrated by a star, and using the MOLAP suggested in this paper - illustrated with a diamond. Both are elevated to the same altitude to improve visibility. In this work $k_2 = 2$ which means that a particle needs to have a weight twice the average weight of all particles in order to be considered. Furthermore $\Omega = 3$ which means that at most $25(5^\circ \cdot 5cm)$ different states are considered³ when evaluating a particular particle. See section 5.2 for a definition of k_2 and Ω .

Finally a test is conducted to illustrate the effect of utilising bootstrapping compared to traditional SMC. In case of standard SMC the tracker was manually initialised to the correct pose 100 frames earlier than the image shown in figure 3. After 100 frames the five best MOLAPs are illustrated in figure 9 with- and without bootstrapping. The five best MOLAPs are illustrated in both a 3D plot and projected into the image. The best MOLAP is found utilising equation 8. The second best is also found using equation 8 but without considering the states in the hypercube (a square in our case) of the best MOLAP, etc.

³Assuming a resolution of 1° for α and $1cm$ for H_z .

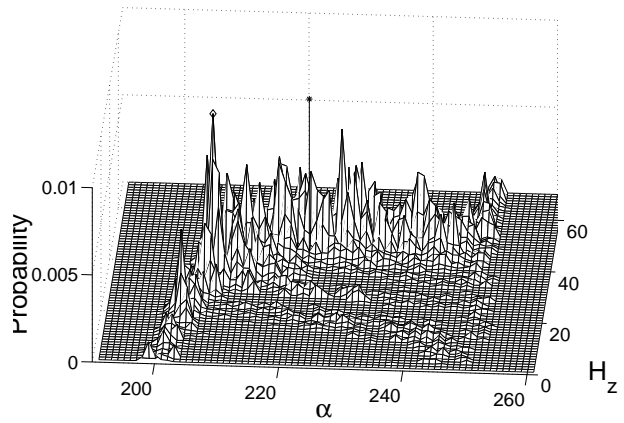
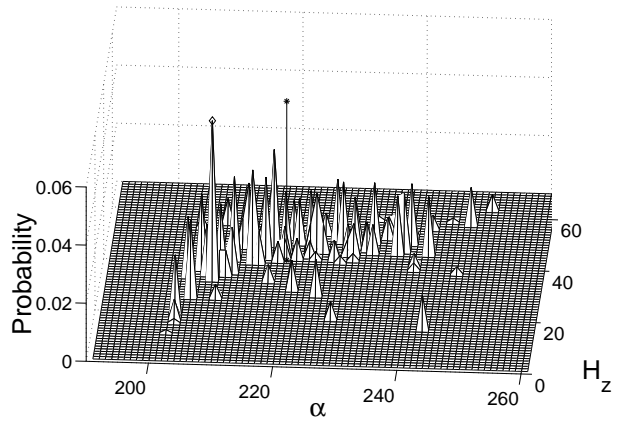
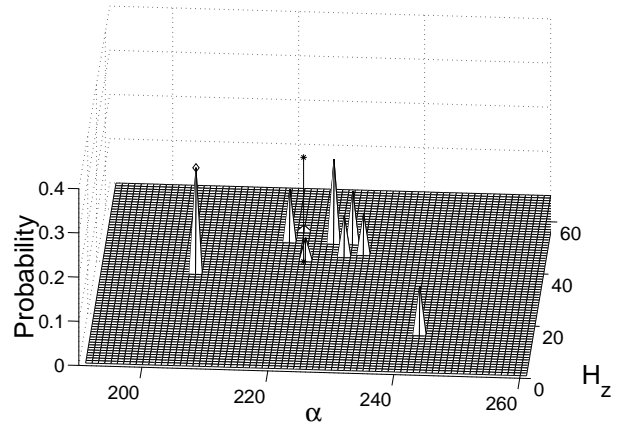


Figure 8: The posterior PDF for $N = 10$, $N = 100$, and $N = 10000$, respectively. The '*' illustrates the current state estimated by the first moment. The '◇' illustrates the current state estimated by the MOLAP.

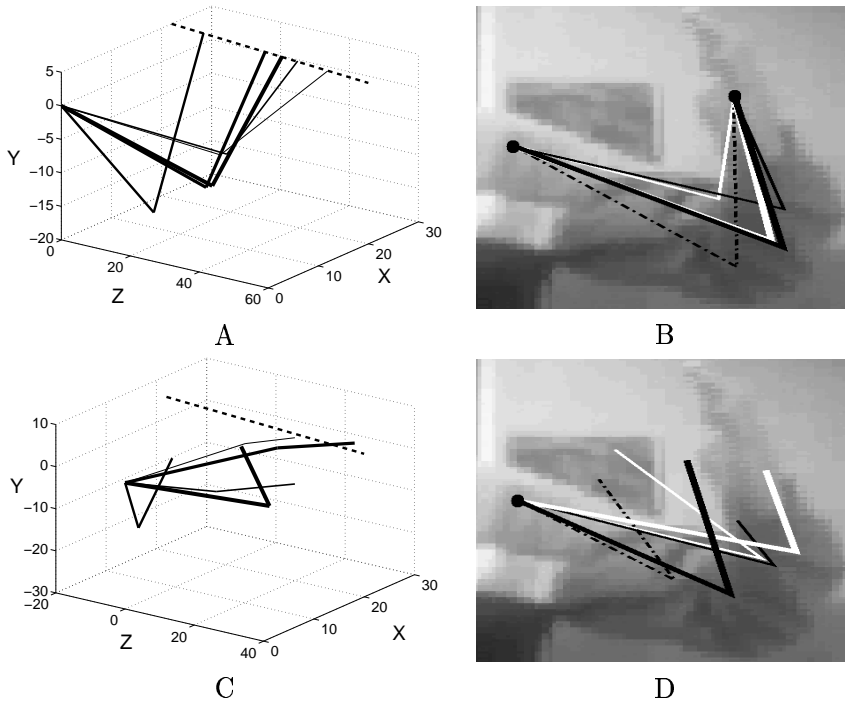


Figure 9: The five most likely configurations of the arm in the image in figure 3.A with bootstrapping (A and B) and without bootstrapping (C and D). For the 3D plots: the thicker the line the higher the likelihood. The dotted line illustrates the camera ray passing through the hand. For the 3D configurations projected into the image: the probability of the lines are in the following order (smallest probability first): thin white, thin black, dash-dotted, thick white, thick black). Note that only the relevant part of image 3.A is shown.

7 Discussion and Conclusion

7.1 Summary

In this paper we have suggested to use distinct image features to bootstrap model-based tracking. Our suggestion is tested in the context of monocular tracking of the 3D pose of a human arm using the SMC algorithm.

First we showed in general terms how to bootstrap the deterministic and stochastic parts utilised in prediction. Then we showed how the position of the hand in the image can be used to bootstrap the state-space representation of the arm, yielding the local screw axis model. Later we showed how the predicted parameters of the model (state-space) can be corrected according to the bootstrapping information and hereby providing a better prediction. The above was implemented in the SMC algorithm where the image measurements were in the form of orientations of the upper- and lower arm, respectively. These were estimated using the dynamic Hough transform (DHT) with a dynamic stop criterion based on the stability of the variance of the variances. Finally we suggested to re-formulate the problem of estimating the MAP to a matter of estimating the MOLAP (most likely a posteriori). A local method for estimating the current state was presented. It takes the multi modal nature of the posterior PDF into account and hereby improving the result.

7.2 Discussion

To justify our methods some tests were conducted. First, the dynamic stop criterion used in the DHT was tested, see figure 6 and 7. The tests clearly show that the variances converge suggesting that the variance of the variances is a solid foundation for the stop criterion. The sub-figures in the second row in figure 6 show the estimated PDFs according to the stop criterion. Assuming the PDFs in the last row to be the "ground truth" a visual comparison reveals that the PDFs estimated by the stop criterion are not identical to the ground truth. Nevertheless, it is evidently that the primary tendencies are kept even though only a fraction of the samples have been applied, hence the dynamic stop criterion is valid.

In our implementation of the bootstrapped SMC algorithm we tried different values of N . In some cases N can be chosen as low as 10 and still producing an accurate estimate of the current state. In general $N = 50$ produces accurate results, but sometimes up to 100 samples are required to ensure an accurate approximation of the posterior PDF and the MOLAP. As the choice of N is not critical we choose $N = 100$. In figure 8 we show an example of a situation where $N = 10$ actually *is* sufficient to estimate the "true" MOLAP according to the MOLAP in the "true" posterior PDF (bottom figure). For $N = 100$ we not only estimate the "true" MOLAP but also have a structure of the posterior PDF similar to that of the "true" posterior PDF. An improvement might be obtained if the value of N is changed in each image according to the current need. One way of controlling N could be to let it depend on the uncertainty in the DHT, i.e., n . Whether this is a sound approach will be investigated in future work.

The standard estimate of the current state using a weighted average is illustrated by a star and the new method (MOLAP) is illustrated with a diamond in figure 8. The method for estimating the current state that has been suggested in this paper might not always find the highest peak. However, it is always located at a (often large) peak with high probabilities in the proximity. The standard estimate of the current state, on the other hand, tends to be located in the centre of the state-space and often at a position where no peaks are present at all, see figure 8. Furthermore, it sometimes happens that the standard estimate of the current state is located in a pruned region. Obviously, this never occurs when using the method suggested in this paper.

In figure 9 the apparent benefits of applying bootstrapping is shown. Even in the case of poorly defined image measurements, as seen in figure 3, our approach provides good results. In images such as the one in figure 3.A the posterior PDF is in general ambiguous. In this particular case a number of correct poses can be found by increasing α as the distance between the hand and camera increases. This tendency can be seen in figure 8. Figure 9 clearly shows that SMC without bootstrapping fails to capture this tendency.

The tendency means that the estimated MOLAP might be incorrect in this particular image. However, due to the ill-posed nature of the problem this will always be the case independently of the chosen tracking framework. The advantage is that in bootstrapped tracking *the* correct peak is virtually always among the largest peaks and will therefore evolve into the next image, hence the tracking approach handles multiple hypotheses.

A future extension is to use bootstrapping in articulated objects with more degrees of freedom, e.g., the entire human body. The bootstrapping information will then be in the form of the position of the two hands, the head, the feet, and possibly other extremities of the human body. In this case the choice of N *will* be an issue. A possible solution might be to apply some of the concepts in [33], that is, only represent the posterior PDF by the 'X' best MOLAPs and a number of samples in the proximity sampled according to the covariances of the MOLAPs.

If the application allows for a delay in the output we can smooth all estimated MOLAPs

over time, e.g., with a Kalman filter, to achieve a more consistent tracking. However, this again introduces the problem of ending up in a location where no peak is present or even worse in a pruned location. Future work will therefore include an investigation of a way to smooth the tracking data based on the r largest peaks in each image instead of only *the* largest peak in each image. One approach could be to use an expression similar to 7 and expand it with the weighted displacements over time to enforce an overall smooth motion on the data.

7.2.1 Related Work

The approach reported in this paper is to some extent comparable to *importance sampling* where the approach is to sample from an *importance function* instead of the prior PDF. The function seeks to concentrate the particles around the peaks corresponding to the different hypotheses and thereby avoiding particles with a low weight. The importance function can be defined in many ways but especially the definitions in [18] and [37] are related to ours. In [18] the additional information is in the form of skin pixels while a texture template (from the head) is used in [37].

Our approach differs in general from importance sampling in the following ways. Importance sampling refines the predictions by concentrating the samples around the peaks where our approach improves the predictions according to current measurements. This means that for instance when sudden motion (or motion not modelled) is present the importance function directs the particles in the correct direction, but will always be limited by the motion model. So, unexpected motion is hard to capture in importance sampling even when the process noise is set very high. In our approach the predictions are corrected or rather carried out with respect to the additional information, hence our approach reacts almost equally well independently of whether the current movement is in accordance with the motion model or not.

The process noise in our approach can in general be low due to the above mentioned and in fact controlled directly by the innovation. Furthermore, we have an immediate relation between the innovation and N which can be used to control N . That is, large innovation indicates large uncertainty, which suggest the need for a large N . This relation is seldom seen in other systems where N therefore is fixed. N has to be set so the algorithm can handle worse case motions, hence it is set much higher than the mean of the required values.

The use of importance sampling in [18] and [37] also allows a continuous tracking even when sudden motion occurs. Our method differs from their methods in the following ways. Our additional information is directly integrated into the actual state-space representation allowing a collapse of the state-space representation. In our case from four parameters into two parameters. Furthermore, our estimation of the process noise also allows a coherent correction of all state-space parameters, where [18] and [37] use their additional information to provide reliable initial samples and to be able to track gross motion, that is, only some of their state parameters are bootstrapped.

Besides relating to importance sampling our approach can also be seen as an attempt to combine the normally distinct approaches of low-level feature tracking and high-level state-space tracking. Since we assume that our low-level tracking is deterministic we do not consider our approach to be a truly multi-cue tracking approach, but rather a high-level tracker approach bootstrapped by low-level information.

Compared to other SMC approaches we calculate the image measurements (in our case the two orientations) for all possible states projected into the image. Normally the image measurements are only calculated for the predicted states. In our case this could have been done by evaluating the sum-of-squared-differences (SSD) between all temporal edge pixels and each predicted state. Which approach is more efficient depends on the number of particles and

the complexity of the SSD and the DHT. As the number of samples in the DHT, n , depends both on the number of temporal edge pixels and their quality n is found dynamically and no general rule can therefore be given. That is, it differs from image to image which approach has less complexity. However, as our approach is seldom seen in SMC methods and can work equally well we chose to implement it and thereby promoting this type of approach.

7.3 Conclusion

In this paper we have suggested to apply bootstrapping to improve performance in model-based tracking. Besides the tests presented above the improvement achieved by this approach can also be understood intuitively. Just imagine the complex nature of the posterior PDF utilising four Euler's angles or the screw axis representation without bootstrapping. Concretely our primary contribution is the concept of applying image measurements from the current frame to improve the state-space representation, the predictions, and the process noise. We have showed the effect of this concept in the context of an SMC method, but the concept is also valid in other tracking frameworks. Furthermore, a contribution is made by suggesting a local method, as opposed to a global method, for estimating the current state. Altogether we therefore conclude that bootstrapped tracking is a promising approach when solving some of the inherent problems in model-based tracking.

References

- [1] A.G. Bharatkumar, K.E. Daigle, M.G. Pandey, Q. Cai, and J.K. Aggarwal. Lower Limb Kinematics of Human Walking with the Medial Axis Transformation. In *Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, Texas, USA, 1994.
- [2] M.J. Black and D.J. Fleet. Probabilistic Detection and Tracking of Motion Discontinuities. In *International Conference on Computer Vision*, Corfu, Greece, September 1999.
- [3] M.J. Black and A.D. Jepson. A Probabilistic Framework for Matching Temporal Trajectories: Condensation-Based Recognition of Gestures and Expressions. In *European Conference on Computer Vision*, Freiburg, Germany, June 2-6 1998.
- [4] A. Blake and M. Isard. *Active Contours*. Springer, 1998.
- [5] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *International Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, June 1998.
- [6] L. Bretzner, I. Laptev, and T. Lindeberg. Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering. In *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, May 20-21 2002.
- [7] T.J. Cham and J.M. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In *Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 23-25 1999.
- [8] Y. Chen, Y. Rui, and T. Huang. Mode-based Multi-Hypothesis Head Tracking Using Parametric Contours. In *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, May 20-21 2002.

- [9] K. Choo and D.J. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In *International Conference on Computer Vision*, Vancouver, Canada, July 9-12 2001.
- [10] L. Davis, V. Philomin, and R. Duraiswami. Tracking Humans from a Moving Platform. In *International Conference on Pattern Recognition*, Barcelona, Spain, September 3-8 2000.
- [11] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, June 13-15 2000.
- [12] A. Doucet, N. Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [13] R. Ginhoux and J.S. Gutmann. Model-Based Object Tracking Using Stereo Vision. In *International Conference on Robotics and Automation*, Seoul, Korea, May 21-26 2001.
- [14] A. Hilton. Towards Model-Based Capture of a Persons Shape, Appearance and Motion. In *International Workshop on Modeling People at ICCV'99*, Corfu, Greece, September 1999.
- [15] M. Isard and A. Blake. A Mixed-state CONDENSATION Tracker with Automatic Model-switching. In *International Conference on Computer Vision*, Bombay, India, January 4-7 1998.
- [16] M. Isard and A. Blake. A Smoothing filter for CONDENSATION. In *European Conference on Computer Vision*, Freiburg, Germany, June 2-6 1998.
- [17] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal on Computer Vision*, 29(1):5-28, 1998.
- [18] M. Isard and A. Blake. ICONDENSATION: Unifying Low-level and High-level Tracking in a Stochastic Framework. In *European Conference on Computer Vision*, Freiburg, Germany, June 2-6 1998.
- [19] G.J. Jang and I.S. Kweon. Robust Object Tracking Using an Adaptive Color Model. In *International Conference on Robotics and Automation*, Seoul, Korea, May 21-26 2001.
- [20] Y. Kameda and M. Minoh. A Human Motion Estimation Method Using 3-Successive Video Frames. In *International Conference on Virtual Systems and Multimedia*, 1996.
- [21] G. Loy, L. Fletcher, N. Apostoloff, and A. Zelinsky. An Adaptive Fusion Architecture for Target Tracking. In *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, May 20-21 2002.
- [22] T.B. Moeslund. Pruning the Possible Configurations of a Human Arm using Kinematic Constraints. Technical Report CVMT 01-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark, 2001.
- [23] T.B. Moeslund. Modelling the Human Arm. Technical Report CVMT 02-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark, 2002.
- [24] T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), 2001.

- [25] T.B. Moeslund and E. Granum. Modelling and estimating the pose of a human arm. *Machine Vision and Applications*, 14(4):237–247, 2003.
- [26] T.B. Moeslund, M. Vittrup, K.S. Pedersen, M.K. Laursen, M.K.D. Sørensen, H. Uhrenfeldt, and E. Granum. Estimating the 3D Shoulder Position using Monocular Vision and a Detailed Shoulder Model. In *International Conference on Imaging Science, Systems, and Technology*, Las Vegas, USA, June 24-27 2002.
- [27] E.J. Ong and S. Gong. Tracking Hybrid 2D-3D Human Models from Multiple Views. In *International Workshop on Modeling People at ICCV'99*, Corfu, Greece, 1999.
- [28] S.M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Wiley Series in Probability and Mathematical Statistics, 1987.
- [29] W. Rungtarityotin and T.E. Starner. Finding Location using Omnidirectional Video on a Wearable Computing Platform. In *International Symposium on Wearable Computers*, Atlanta, Georgia, USA, October 18-21 2000.
- [30] P. Sangi, J. Heikkilä, and O. Silven. Extracting Motion Components from Image Sequences using Particle Filters. In *The 12th Scandinavian Conference on Image Analysis*, Bergen, Norway, 2001.
- [31] H. Sidenbladh, M.J. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, Copenhagen, Denmark, 2002.
- [32] H. Sidenbladh, F. De la Torre, and M.J. Black. A Framework for Modeling the Appearance of 3D Articulated Figures. In *The fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, March 2000.
- [33] C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3D Body Tracking. In *Computer Vision and Pattern Recognition*, Kauai Marriott, Hawaii, December 9-14 2001.
- [34] D. Tolani and N.I. Badler. Real-Time Inverse Kinematics of the Human Arm. *Presence*, 5(4), 1996.
- [35] D. Tolani, A. Goswami, and N.I. Badler. Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graphical Models*, 62(5), 2000.
- [36] V.M. Zatsiorsky. *Kinematics of Human Motion*. Champaign, IL: Human Kinetics, 1998.
- [37] Z. Zeng and S. Ma. Head Tracking by Active Particle Filtering. In *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, May 20-21 2002.
- [38] S. Zhou, V. Krueger, and R. Chellappa. Face Recognition from Video: A CONDENSATION Approach. In *International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, May 20-21 2002.