Towards Human Motion Capture from a Camera mounted on a mobile Robot

Paulo Menezes[†], Frédéric Lerasle[‡¶], Jorge Dias[†]

[†] *ISR-University of Coimbra, Polo II, 3030-290 Coimbra, Portugal*
[‡] *CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse Cedex 4, France*
[¶] *Université de Toulouse; UPS, INSA, INP, ISAE; UT1, UTM, LAAS; F-31077 Toulouse Cedex 4, France*
Corresponding author: F.Lerasle, Tel: (+33) 05 61 33 69 61, E-mail: lerasle@laas.fr

1 Introduction and framework

A major challenge of Robotics is undoubtedly the assistant robot, with the perspective having such an autonomous mobile platform to serve humans in their daily life. Embedding human motion capture (HMC) systems, based on conventional cameras, on the robot would give it the ability to (i) act in a social and human aware way and, (ii) interact with humans in a natural and rich way. While the Vision community proposes a plethora of remarkable works on HMC from ceiling-mounted and wide-angle cameras (see a survey in [24,30]), only a few of them address robotic applications [2,19]. 3D tracking from a mobile platform is arguably a challenging task, which imposes several requirements. First, the embedded camera covers a narrow field of view comparatively to ambient multi-ocular systems. As the robot's operation takes place on a wide variety of environmental conditions, background modelling techniques techniques [5,34,35] are precluded, while the tracker is inevitably faced with ambiguous data. In these situations, several hypotheses must be handled simultaneously, and a robust integration of multiple visual cues is required. Finally, on-board processing power is limited, thus care must be taken to

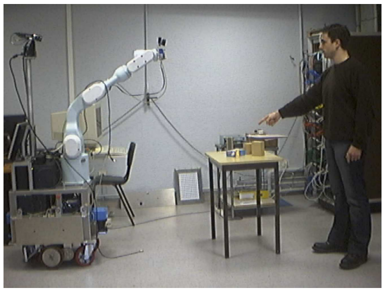


Figure 1. An interactive task requiring HMC.

design computationally efficient algorithms. Like many researchers of the Vision community, we aim at investigating markerless HMC systems based on embedded vision techniques. In brief, our framework differs clearly from (i) pedestrian detection [10,11,21] which aims at estimate the global person pose, and (ii) HMC from multiple wide-angle cameras [13,35,40] in home environment surveillance applications.

Most of the existing approaches in our context have concentrated on 3D articulated models of the tracked human limbs in order to make the problem more tractable. These approaches essentially differ in the associated data processing (namely 3D reconstruction *versus* appearance based approaches) and the estimation framework (namely deterministic *vs.* Bayesian approaches). Reconstruction-based approaches try to fit the model to the 3D-point cloud issued from 3D-sensor system *e.g.* a stereo head [3,40,42] or a Swiss Ranger [19]. Besides, the appearance-based approaches infer the model configuration from its projections in the video stream [5,20,31,35,41]. These strategies enable the use of abundant appearance information which can be extracted from the image contents. We perform the integration of all those visual cues in a rigorous probabilistic way, since we consider a Bayesian formulation for posture estimation. For this, we use the particle filtering framework [7] which has become popular for tracking, since it was pioneered by Isard *et al.* in the form of CONDENSATION [15]. As it makes no restrictive assumptions on the probability distributions to be propagated, particle filtering is well suited to the above robotic requirements.

The main drawback of using conventional particles filters remains on the number of particles that are required to efficiently sample the high dimensional state space. Search space decomposition techniques [5] undoubtedly enable to tackle this problem, yet global view strategies[1] are often favoured to determine the correct human posture. To limit the required number of particles (and so the computational load) we need an alternative to the widely used CONDENSATION algorithm for HMC [2,20,33] that enables the design of a better proposal distribution. Indeed, samples in the CONDENSATION scheme are predicted blindly *i.e.* thanks to an importance function which depends only on dynamic models. The latter, typically based on Gaussian random walk, are quite diffuse and poor, because inter-frame human motions are difficult to characterise. Consequently, very large particle sets are required in order to achieve acceptable performance. Besides, alternative sampling schemes have been suggested to steer sampling towards state space regions of high likelihood by incorporating the current observation. Unscented particle filters [32] have shown to outperform the CONDENSATION scheme but they still constrain the importance function to be Gaussian, which leads to inefficient sampling in the presence of multi-modal distributions. In the same vein, the ICONDENSATION framework [16], by positioning samples according to visual detectors *i.e.* independently of the past state, addresses also this problem but this may lead to be con-

[1] *i.e.* involving all the human parts in a single step.

tradicting with the process history and so to an inefficient filtering (see [39] for explanations).

Recent approaches like[9] use an approach that mixes the principles of particle filtering with global optimisation capabilities of simulated annealing with good results, but still requiring 4 cameras and about 61 seconds of processing time per frame.

Pitt *et al.* in [29] have suggested the so-called AUXILIARY scheme which is shown to overcome this problem and to propose an optimal importance function (see [8] for theoretical developments). The proposal distribution is an appropriately defined mixture that depends both on the past state and the current observation. This genuine strategy has been surprisingly seldom exploited for tracking purpose [12,18]. Conventionally, this strategy involves the evaluations of data-driven likelihoods both in the prediction and weighting stages. Consequently, the computational burden when applied to visual tracking may be pretty high [39], especially for tricky tracking in high state space dimension. Our opposite view is to differentiate clearly the data-driven likelihoods involved in these two stages. For the proposal density, we propose a basic and coarse likelihood in order to place samples in the relevant regions of the state space without greatly increasing the global time consumption of the filter. On the contrary, the likelihood function involved in the weighting stage is more elaborated as it incorporates several visual cues and geometric constraints with appropriate degrees of adaptivity depending on the human posture and the environmental context encountered by the robot while navigating during in indoor human-centred environments.

The rest of this paper is organised as follows. Section 2 focuses on the modelling of the 3D structure, and presents a variant from the generic technique for projecting and handling self-occlusions between parts. Section 3 briefly outlines the well-known particle filtering formalism, its limitations, and the auxiliary scheme which permits a multilevel integration of multiple cues. Section 4 presents the different likelihoods involved in our AUXILIARY scheme. Implementation and experiments on two-arm gestures tracking are presented in section 5. Section 6 summarises our contribution and opens the discussion for future extensions.

2 From the 3D model to its appearance

Performing 3D tracking from a single camera requires the use of a model that compensates for the lack of depth information. Although recent works, like [9], use polygonal meshes to create hight fidelity models, we use simpler models based on coarse truncated quadrics which are perfectly adequate for our purpose. Quadrics are quite popular geometric primitives for use in human body tracking [3,4,37]. This is due to fact that are easily handled, they can be combined to create complex shapes, and their projections are conic sections that can be obtained in closed form. This section starts by recalling some basics on quadrics and their projective models, followed by the presentation of the model, it ends with the details of generation of the appearance model which is used in the measuring stage of the estimation process. One could argue that moderns graphics cards can be used to obtain the model projection, solving all the visibility issues that we address here. Although this hypothesis was considered we found that would introduce another level of complexity for the following reasons: 1) this would require the creation of OpenGL projective cameras that correspond exactly to our physical camera. This would introduce some problems especially for low cost cameras that normally present considerable deviations from the ideal camera in terms of the "principal point" (the intersection of the lens axis with the retina) being away from the image centre and skew factors frequently different from 0; 2) the second problem comes from the fact that the model projection obtained with graphics cards are textures and not lists of line segments as required by our approach.

2.1 Basics on projective geometry modelling

A quadratic surface or quadric $\mathcal{Q}$ is a second degree implicit surface in 3D space. It can be represented in vectorial form using homogeneous coordinates as $\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0,\ \forall\mathbf{X} \in \mathcal{Q}$, where $\mathbf{X}$ is a 4×1 vector and $\mathbf{Q}$ is a 4×4 symmetric matrix. To employ quadrics for modelling more general shapes, it is necessary to truncate them, and eventually combine them, in our case we used truncated cylinders and cones to approximate the upper limbs shapes. For a given quadric $\mathcal{Q}$, the truncated counterpart is defined by the set of points $\mathbf{X}$ that verify

$$\begin{cases} \mathbf{X}^T\mathbf{Q}\mathbf{X} = 0 \\ \mathbf{X}^T\Pi\mathbf{X} \leq 0 \end{cases} \tag{1}$$

where $\mathbf{\Pi}$ is a matrix representing the pair of clipping planes which delimit the quadric $\mathcal{Q}$. Finally, it should be noted that a given quadric $\mathcal{Q}^{'}$ resulting from the application of an Euclidian transformation $\mathbf{H}$ to the points of the surface $\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0$ is $\mathbf{X}^{'T}\mathbf{Q}^{'}\mathbf{X}^{'} = 0$ where $\mathbf{X}^{'} = \mathbf{H}\mathbf{X}$ and $\mathbf{Q}^{'} = \mathbf{H}^{-T}\mathbf{Q}\mathbf{H}^{-1}$.

2.2 Description of the human limbs model

Considering N_p parts in the 3D model, the latter is built using a set of rigid truncated quadrics $\mathcal{Q}_i$ $(i \in 1, \ldots, N_p)$ to represent approximatively the shape of the human limbs. The truncated quadrics are connected between them by articulations where each one can contain one or more degrees of freedom (DOF). In the current work, eight DOF, *i.e.* four per arm (3 rotations on the shoulder and 1 rotation on the elbow), are actually tracked by this system. The arm ends, as well as their joints are here represented by spheres in order to improve the visual effect, but they are not being considered as model parts for tracking purposes.

2.3 Generation of the model projection

2.3.1 Projection of a quadric

Let us start with the projection of a quadric in a normalized camera. The associated projection matrix is $\mathbf{P} = [\mathbb{I}_{3\times 2}|\mathbf{0}_{3\times 1}]$. Considering a pinhole camera model, the camera center and an image point $\mathbf{x}$ define a projective ray which contains the points given by $\mathbf{X} = [\mathbf{x}\ s]^T$, as illustrated on Figure 2. The depth of a 3D point, situated on this ray, is determined by the scalar s. The expression of the quadric $\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0$ can then be rewritten as

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + 2s\mathbf{b}^T\mathbf{x} + s^2c = 0, \ \text{ where } \ \mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix}. \tag{2}$$

This expression can be considered as an equation of second degree in s. Then, when the ray represented by $\mathbf{X}(s)$ is tangent to $\mathcal{Q}$, there is a single solution for equation (2) *i.e.* its discriminant is zero, so: $\mathbf{x}^T(\mathbf{b}\mathbf{b}^T - c\mathbf{A})\mathbf{x} = 0$. This expression corresponds to a conic $\mathcal{C}$ in the image plane, with $\mathbf{C} = c\mathbf{A} - \mathbf{b}\mathbf{b}^T$, which therefore corresponds to the projection of the quadric $\mathcal{Q}$. For any $\mathbf{x}$ belonging to $\mathcal{C}$, the corresponding 3D point $\mathbf{X}$ is fully defined by finding s_0 which is given by $s_0 = -\mathbf{b}^T\mathbf{x}/c$. This formula can be extended to arbitrary projective camera with $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$. Defining a 4×4 matrix $\mathbf{H}$ such that $\mathbf{P}\mathbf{H} = [\mathbb{I}|\mathbf{0}]$, and then $\mathbf{x} = [\mathbb{I}|\mathbf{0}]\mathbf{H}^{-1}\mathbf{X}$.

2.3.2 Projection of our model's components

Each of the quadrics, composing the model, is said to be degenerate because its matrix, $\mathbf{Q}$, is singular. The image of such a degenerated quadric, obtained by projective projection, is a degenerated conic. Being our model composed of cones and cylinders, and depending on the point of view, its image can be a pair of parallel lines, a pair of concurrent lines, or even a single point. These three cases are easy to identify if matrix $\mathbf{C}$ is diagonal. A diagonal matrix $\mathbf{C}$ corresponds to a conic aligned with the x- and y-axis and centered at the origin. To achieve this diagonalization, Stenger *et al.* in [36] use eigen-decomposition while we propose a more intuitive method. Our strategy is to define analytically a 3×3 matrix $\mathbf{B}$ that brings the conic into this particular configuration, thence diagonalizing the $\mathbf{C}$ matrix.

For the more general cases, that correspond to having two projected lines, we can easily determine two points (denoted $\mathbf{x}_1$ and $\mathbf{x}_2$) on each of the lines. These rendered points are then transformed back into their original configuration through transformation $\mathbf{B}^{-1}$, characterising then each projected line l in the original frame.

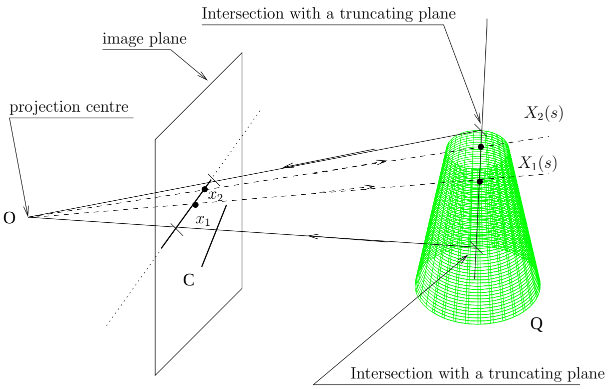

Intersection with a truncating plane
image plane
projection centre
O
x_2
x_1
C
$X_2(s)$
$X_1(s)$
Q
Intersection with a truncating plane

Figure 2. Truncating projected segments.

The next step is to truncate the projections of the cones. Each 3D line segment L, on the quadrics surface, corresponding to a projected line l segment, is characterised by two points $\mathbf{X}_i$ which are associated with image points $\mathbf{x}_i$. These 3D points are computed by $\mathbf{X}_i = [\mathbf{x}_i \ - \frac{\mathbf{b}^T\mathbf{x}_i}{c}]^T$ where $i = 1, 2$, where $\mathbf{b}$ and c are blocks of the quadric's matrix. The 3D line defined by these two points corresponds to the visibility frontier of the quadric surface. A generic point on this line is then given by

$$\mathbf{X} = \mathbf{X}_1 + \lambda\mathbf{X}_2, \ \forall\, \mathbf{X} \in L. \tag{3}$$

The intersections that happen between this line and the two clipping planes are given by $(\mathbf{X}_1 + \lambda\mathbf{X}_2)^T\Pi(\mathbf{X}_1 + \lambda\mathbf{X}_2) = 0$. This is in fact a second degree equation in λ as all the other involved points are known. Solving this equation gives two values for λ that, once substituted back into (3), produce the required 3D (intersection) points. These ones, are then re-projected onto the image plane to obtain the extremities of the line segments that correspond to the cone's projection. This procedure is illustrated on Figure 2.

2.3.3 Self-occlusion handling

The final step is the handling of self-occlusion. There are several algorithms available in the literature to manage the hidden parts of a model [25,38]. Most of them are computationally heavy or inadequate for the current problem. One example is a recent algorithm presented in [36] that, although being quite adequate to the problem, presents a complexity which depends both on the size of the projected parts and on the required precision. In the current work, the use of quadrics of conic or cylindric type enables the use of an method whose complexity depends only on the number of projected parts and not on their size or precision. This algorithm, that will be described hereafter, requires, consequently, less computational power than the former ones.

The algorithm starts with the computation of all the strict intersections amongst the whole set of projected segments. Strict intersection between two segments is

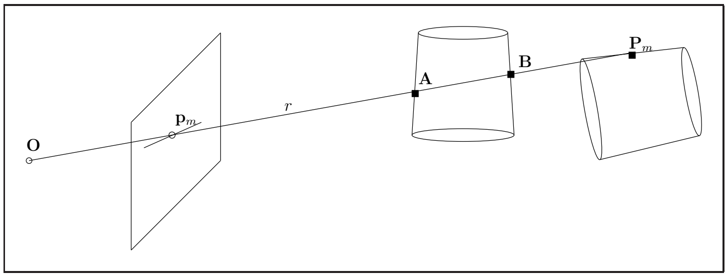


Figure 3. Hidden segment testing.

defined as the case where two segments intersect and the intersection point is not an extremity of either segment. The computation of these intersections can benefit from the application of the sweeping line algorithm [26], especially if the number of projected segments is large, as this method presents a complexity that is linear with the number of segments while the usual brute force method presents a quadratic one.

	Time consumption (sec)
Quadrics transformation	1.38×10^{-4}
Contours projection	3.47×10^{-4}
Hidden parts removal	5.25×10^{-4}
Total	1.01×10^{-3}

Table 1
Time consumption during model projection.

For each intersection point, the two implicated segments are sectioned at this point. At the end there will be a list of (smaller) segments that do not exhibit any strict intersection between them. The next step is to use the middle point, $\mathbf{x}_m$, of each segment, and the camera center to define a projective line. Lets recall that every point on this line projects onto $\mathbf{x}_m$, as show by figure 3. The computation of the set of intersections between this line and the whole set of quadrics enable us to verify if the 3D point that originated $\mathbf{x}_m$ is in front of all the other quadrics or is hidden by any of them, the same happening to the segment that contains the point.

For each of the segments of the new set, its middle point p_m is computed, which in conjunction with $\mathbf{O}$ (the camera centre) defines a projective ray. A search is then performed to find possible intersections between this projective ray and any of the quadrics that compose the 3D structure. The intersection points are then ordered using their distance to the camera centre. The segment is then marked as visible or invisible, whether the first point of intersection in the list belongs or not to the quadric that originated it (through projection).

Figure 3 illustrates this idea, $\mathbf{p}_m$ is the middle point of an image segment, r is the projective line that passes by $\mathbf{p}_m$ and $\mathbf{O}$, and $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{P}_m$ the intersection points with the two quadrics. From this test it results that the projected segment is not visible as both the points A and B are closer to O than P which is the one that is on the surface of the cone that originated this segment.

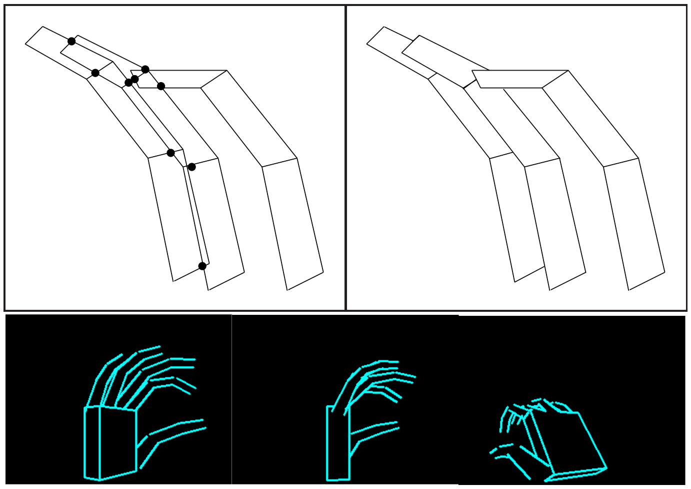


Figure 4. Top Row: Illustration of a projection before (left) and after (right) hidden line removal. Bottom tow: examples of the application of the algorithm to a hand model.

This is done by defining that any point on the projective ray that passes through p_m will have projective coordinates of the form

$$\mathbf{X}_p = \begin{bmatrix} \mathbf{p}_m \\ s \end{bmatrix}$$

where s is a scalar. Then replacing $\mathbf{X}_p$ in the equation of each quadric, a second degree equation in s will be obtained. Then for each obtained s the resulting point $\mathbf{X}_p$ is tested to see if it is situated between the corresponding cone truncating planes, in other words, if the point verify the inequality $\mathbf{X}_p^T \mathbf{\Pi} \mathbf{X}_p \geq 0$ then it belongs to the truncated cone, otherwise it belongs to the cone but it is out of the zone delimited by the two clipping planes.

It should be noted that comparing distances between the intersection points can be done by just comparing the respective perspective coordinates, s, whereas the larger this value, the smaller the distance from the considered point to the camera centre.

Contrary to other methods that test the visibility of every projected point [36], a single test on the segment's middle point is enough to infer about the visibility of a segment.

Figure 4 (tow row) illustrates the intersection points between projections of the segments and the result after hidden segments removal. The bottom row of of the same figure shows the results of the application of this method to the projections of a hand model. A simple performance evaluation was done when handling a 3D model composed of the trunk/two arms and so 5 quadrics and 10 truncating planes. The time consumption is reported in Table 1 for the following operations: transform the whole set of quadrics (cones and truncating cones), project them, and charac-

```
$[\{x_k^{(i)}, w_k^{(i)}\}]_{i=1}^N$= SIR($[\{x_{k-1}^{(i)}, w_{k-1}^{(i)},\}]_{i=1}^N, z_k$)
IF $k = 0$, THEN Draw $x_0^{(1)}, \dots, x_0^{(i)}, \dots, x_0^{(N)}$ i.i.d. according to $p(x_0)$, and set $w_0^{(i)} = \frac{1}{N}$ END IF
IF $k \geq 1$ THEN {—$[\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}]_{i=1}^N$ being a particle description of $p(x_{k-1}|z_{1:k-1})$—}
  FOR $i = 1, \dots, N$, DO
    "Propagate" the particle $x_{k-1}^{(i)}$ by independently sampling $x_k^{(i)} \sim q(x_k|x_{k-1}^{(i)}, z_k)$
    Update the weight $w_k^{(i)}$ associated to $x_k^{(i)}$ according to $w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)}$,
    prior to a normalization step so that $\sum_i w_k^{(i)} = 1$
  END FOR
  Compute the conditional mean of any function of $x_k$, e.g. the MMSE estimate $\mathrm{E}_{p(x_k|z_{1:k})}[x_k]$, from the approximation $\sum_{i=1}^N w_k^{(i)} \delta(x_k - x_k^{(i)})$ of the posterior $p(x_k|z_{1:k})$
  At any time or depending on an "efficiency" criterion, resample the description $[\{x_k^{(i)}, w_k^{(i)}\}]_{i=1}^N$ of $p(x_k|z_{1:k})$ into the equivalent evenly weighted particles set $[\{x_k^{(s^{(i)})}, \frac{1}{N}\}]_{i=1}^N$, by sampling in $\{1, \dots, N\}$ the indexes $s^{(1)}, \dots, s^{(N)}$ according to $P(s^{(i)} = j) = w_k^{(j)}$; set $x_k^{(i)}$ and $w_k^{(i)}$ with $x_k^{(s^{(i)})}$ and $\frac{1}{N}$
END IF
```

Table 2
Generic particle filtering algorithm (SIR)

terise their occluding contours. To stress the performance gain relatively to other methods, it should be noted that this one only performs a visibility test for a single point on each projected segment whereas the method presented in [36] has to perform the visibility test for each projected point. Therefore to computing time for the ”hidden parts removal” shown on table 1 corresponds to the verification of the visibility a small number of points for all projected parts. This time should increase linearly with the number of points verified

3 Particle filtering algorithms for data fusion

3.1 Basics strategies and associated limitations

Particle filters are sequential Monte Carlo simulation methods for the state vector estimation of any Markovian dynamic system subject to possibly non-Gaussian random inputs [1]. Their aim is to recursively approximate the posterior density function (pdf) $p(x_k|z_{1:k})$ of the state vector x_k at time k conditioned on the set of measurements $z_{1:k} = z_1, \dots, z_k$. A linear point-mass combination

$$p(x_k|z_{1:k}) \approx \sum_i w_k^{(i)} \delta(x_k - x_k^{(i)}), \ \sum_{i=1}^N w_k^{(i)} = 1. \qquad (4)$$

is determined - with $\delta(.)$ the Dirac distribution - which expresses the selection of a value - or “particle” - $x_k^{(i)}$ with probability - or “weight” - $w_k^{(i)}, i = 1, \dots, N$. An approximation of the conditional expectation of any function of x_k, such as the minimum mean square error (MMSE) estimate $E_{p(x_k|z_{1:k})}[x_k]$, then follows.

The generic particle filter - or “Sampling Importance Resampling” SIR is shown on

Table (2). The particles $x_k^{(i)}$ evolve stochastically over the time, being sampled from an importance density $q(.)$ which aims at adaptively exploring "relevant" areas of the state space. Their weights $w_k^{(i)}$ are updated thanks to $p(x_k^{(i)}|x_{k-1}^{(i)})$ and $p(z_k|x_k^{(i)})$, resp. the state dynamics and measurement functions, so as to guarantee the consistency of the approximation (4). In order to limit the degeneracy phenomenon, which says that after few instants all but one particle weights tend to zero, step 8 inserts a resampling process. Another solution to limit this effect in addition to re-sampling, is the choice of a good importance density.

The $\mathrm{CONDENSATION}$ algorithm is instanced from the SIR as $q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k^{(i)}|x_{k-1}^{(i)})$. Another difference relative to the SIR algorithm presented is that the re-sampling step 8 is applied on every cycle. Resampling by itself cannot efficiently limit the degeneracy phenomenon as the state-space is blindly explored without any knowledge of the observations. On the other side, the $\mathrm{ICONDENSATION}$ algorithm [16], consider importance density $q(.)$ which classically relates to importance function $\pi(x_k^{(i)}|z_k)$ defined from the current image. However, if a particle drawn exclusively from the image is inconsistent with its predecessor in terms of state dynamics, the update formula leads to a small weight [39].

3.2 Towards the "optimal" case: the Auxiliary Particle Filter

It can be shown [7] that the most efficient recursive scheme, *i.e.* the one which best limits the degeneracy phenomenon, must define $q^*(x_k|x_{k-1}^{(i)}, z_k) \triangleq p(x_k|x_{k-1}^{(i)}, z_k)$ and thus $w_k^{*(i)} \propto w_{k-1}^{*(i)} p(z_k|x_{k-1}^{(i)})$ in the SIR algorithm Table 2. This "optimal" strategy noticeably affects each particle $x_k^{(i)}$ a weight $w_k^{*(i)}$ which solely depends on $x_{k-1}^{(i)}$, $w_{k-1}^{*(i)}$ and z_k – through the *predictive likelihood* $p(z_k|x_{k-1}^{(i)})$ – so that $w_k^{*(i)}$ can be computed prior to drawing $x_k^{(i)}$. Further, to enhance the algorithm efficiency, the resampling step can be shifted just before the "propagation" through the optimal importance function $q^*(x_k|x_{k-1}^{(i)}, z_k)$ of the weighted particles set $[\{x_{k-1}^{(i)}, w_k^{*(i)}\}]_{i=1}^N$, which in fact represents the smoother pdf $p(x_{k-1}|z_{1:k})$. Unfortunately, except in very particular cases, the above formulae are of limited practical interest, for it is often impossible to sample from $p(x_k|x_{k-1}^{(i)}, z_k)$ nor to compute $p(z_k|x_{k-1}^{(i)}) = \int p(z_k|x_k)p(x_k|x_{k-1}^{(i)})\mathrm{d}x_k$.

Still, it remains possible to mimic the optimal strategy even if an importance function $\pi(x_k|x_{k-1}^{(i)}, z_k)$ is defined instead of $p(x_k|x_{k-1}^{(i)}, z_k)$ and if only an approximation $\hat{p}(z_k|x_{k-1}^{(i)})$ of the predictive likelihood $p(z_k|x_{k-1}^{(i)})$ can be computed from the current measurement z_k. First, an *auxiliary weight* $\lambda_k^{(i)} \propto w_{k-1}^{(i)}\hat{p}(z_k|x_{k-1}^{(i)})$ is associated to each particle $x_{k-1}^{(i)}$. Then, the approximation $[\{x_{k-1}^{(i)}, \lambda_k^{(i)}\}]_{i=1}^N$ of $p(x_{k-1}|z_{1:k})$ is resampled into the evenly weighted set $[\{x_{k-1}^{(s^{(i)})}, \frac{1}{N}\}]_{i=1}^N$, which is further "prop-

```
$[\{x_k^{(i)}, w_k^{(i)}\}]_{i=1}^N = \mathrm{AUXILIARY}([\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}]_{i=1}^N, z_k)$

IF $k = 0$, THEN Draw $x_0^{(1)}, \ldots, x_0^{(i)}, \ldots, x_0^{(N)}$ i.i.d. according to $p(x_0)$, and set $w_0^{(i)} = \frac{1}{N}$ END IF
IF $k \geq 1$ THEN {—$[\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}]_{i=1}^N$ being a particle description of $p(x_{k-1}|z_{1:k-1})$—}
  FOR $i = 1, \ldots, N$, DO
    From the approximation $\hat{p}(z_k|x_{k-1}^{(i)}) = p(z_k|\mu_k^{(i)})$ –e.g. with $\mu_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$ or $\mu_k^{(i)} = \mathrm{E}_{p(x_k|x_{k-1}^{(i)})}[x_k]$–, compute the auxiliary weights $\lambda_k^{(i)} \propto w_{k-1}^{(i)} \hat{p}(z_k|x_{k-1}^{(i)})$, prior to a normalization step so that $\sum_i \lambda_k^{(i)} = 1$
  END FOR
  Resample $[\{x_{k-1}^{(i)}, \lambda_k^{(i)}\}]_{i=1}^N$ –or, equivalently, sample in $\{1, \ldots, N\}$ the indexes $s^{(1)}, \ldots, s^{(N)}$ of the particles at time $k-1$ according to $P(s^{(i)} = j) = \lambda_k^{(j)}$– in order to get the equivalent evenly weighted particles set $[\{x_{k-1}^{(s^{(i)})}, \frac{1}{N}\}]_{i=1}^N$ ; both $\sum_{i=1}^N \lambda_k^{(i)} \delta(x_{k-1} - x_{k-1}^{(i)})$ and $\frac{1}{N} \sum_{i=1}^N \delta(x_{k-1} - x_{k-1}^{(s^{(i)})})$ approximate the smoothing pdf $p(x_{k-1}|z_{1:k})$
  FOR $i = 1, \ldots, N$, DO
    “Propagate” the particles by independently drawing $x_k^{(i)} \sim p(x_k|x_{k-1}^{(s^{(i)})})$
    Update the weights, prior to their normalisation, by setting $w_k^{(i)} \propto \frac{p(z_k|x_k^{(i)})}{\hat{p}(z_k|x_{k-1}^{(s^{(i)})})} = \frac{p(z_k|x_k^{(i)})}{p(z_k|\mu_k^{(s^{(i)})})}$
    Compute the conditional mean of any function of $x_k$, e.g. the MMSE estimate $\mathrm{E}_{p(x_k|z_{1:k})}[x_k]$, from the approximation $\sum_{i=1}^N w_k^{(i)} \delta(x_k - x_k^{(i)})$ of the posterior $p(x_k|z_{1:k})$
  END FOR
END IF
```

Table 3
Auxiliary Particle Filter (AUXILIARY)

agated” until time k through $\pi(x_k|x_{k-1}^{(s^{(i)})}, z_k)$. Finally, the weights of the resulting particles $x_k^{(i)}$ must be corrected in order to take into account of the “distance” between $\lambda_k^{(i)}$ and $w_k^{*(i)}$, as well as of the dissimilarity between the selected and optimal importance functions $\pi(x_k^{(i)}|x_{k-1}^{(s^{(i)})}, z_k)$ and $p(x_k^{(i)}|x_{k-1}^{(s^{(i)})}, z_k)$. To this end, one must set

$$w_k^{(i)} \propto \frac{p(z_k|x_k^{(i)}) p(x_k^{(i)}|x_{k-1}^{(s^{(i)})})}{\hat{p}(z_k|x_{k-1}^{(s^{(i)})}) \pi(x_k^{(i)}|x_{k-1}^{(s^{(i)})}, z_k)}$$

. The “Auxiliary Particle Filter” (AUXILIARY) was developed in this vein by [29] contemporarily to ICONDENSATION. It approximates the predictive likelihood by $\hat{p}(z_k|x_{k-1}^{(i)}) = p(z_k|\mu_k^{(i)})$, where $\mu_k^{(i)}$ characterizes the distribution of x_k conditioned on $x_{k-1}^{(i)}$ –*e.g.* $\mu_k^{(i)} = \mathrm{E}_{p(x_k|x_{k-1}^{(i)})}[x_k]$ or $\mu_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$–, and its importance function follows the system dynamics, *i.e.* $\pi(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)})$, see Table 3. The particles cloud can thus be steered towards relevant areas of the state space. In the visual tracking context, the approximate predictive likelihood can rely on visual cues which are different from those involved in the computation of the “final-stage” likelihoods $p(z_k|x_k^{(i)})$.

In practice, the AUXILIARY scheme runs slightly slower than the CONDENSATION as we need to perform two weighted bootstraps rather than one. However, the improvement in sampling will usually dominate these small effects. By making proposals which have high conditional likelihoods, we reduce the costs of sampling many times from particles which have very low likelihoods and so will not be resampled at the second process stage. This improves the statistical efficiency of the sampling procedure and means that we can reduce substantially the particles number.

4 Likelihoods entailed in the $\mathrm{AUXILIARY}$ scheme

The next subsections describe the different likelihoods entailed in the proposal distribution and in the measurement function. The latter is based on multiple visual cues and model priors to encode physical properties of the 3D model.

4.1 Predictive likelihood for the proposal distribution

The particles sampled in step 6 (Table 3) are drawn from a likelihood which encodes the similarity between skin-like image ROIs and virtual patches related to the two hands projection given a model configuration. Let h^c_{skin} and h^c_μ be two N_{bi}-bin normalized histograms in channel $c \in \{R, G, B\}$, respectively corresponding to a skin color distribution and to a region B_μ parameterized by the state μ. The colour likelihood $p(z|\mu)$ in Table 3 must favor candidate colour histograms $h^c_{\mu,p}$ for the two hands close to the skin colour distribution h^c_{skin} *i.e.*

$$p(z|\mu) \propto \exp\left(-\frac{D^2}{2\sigma_c^2}\right),\ D = \frac{1}{2}\sum_c \sum_{p=1}^{2} D_B(h^c_{\mu,p}, h^c_{skin}), \tag{5}$$

provided that D_B terms the Bhattacharyya distance [27] between the two histograms $h^c_{\mu,p}$ and h^c_{skin} and σ_c is the standard deviation being determined *a priori*. Training images from the Compaq database [17] enables to model the skin colour distribution h^c_{skin} dedicated to the hands.

4.2 Measurement sub-functions

1. Shape cue: using this cue requires the 3D model projection with hidden parts removing. The shape-based likelihood $p(z^S|x)$ is computed using the sum of the squared distances between model points and the nearest image edges [15]. The measurement points are chosen to be uniformly distributed along the model projected segments. In our implementation, the edge image is converted into a Distance Transform image, noted I_{DT}, which is used to approximate the distance values. The advantage of matching our model contours against a DT image rather than using directly the edges image is that the resulting similarity measure is a smoother function of the model parameters. In addition, the DT image reduces the involved computations as it needs to be generated only once whatever the number of particles involved in the filter. The likelihood $p(z^S|x)$ is given by

$$p(z^S|x) \propto \exp\left(-\lambda_s \frac{D^2}{2\sigma_s^2}\right),\ D = \frac{1}{N_p}\sum_{j=0}^{N_p} I_{DT}(j), \tag{6}$$

where j indexes the N_p model points uniformly distributed along each visible model projected segments, $I_{DT}(j)$ is the associated value in the DT image, σ_s is the standard deviation being determined *a priori*, and λ_s is a weighting factor discussed later.

2. Motion cue: considering a static camera, it is highly possible that the targeted subject be moving, at least intermittently in some H/R situations. To cope with background clutter, we thus favour the moving edges by combining motion and shape cues into the definition of the likelihood $p(z^{MS}|x)$ of each particle x. This is accomplished by using two DT images, noted I_{DT} and I'_{DT}, where the new one is obtained by filtering out the static edges, based on $\vec{f}(z(j))$ the conventional optical flow vector at pixel $z(j)$ assuming the constant brightness assumption *i.e.* considering that the brightness of a pixel will not change between two successive image frames [14]. The likelihood $p(z^{MS}|x)$ has a form similar to (6) with associated parameters σ_m and λ_m, provided that D is given by

$$D = \frac{1}{N_p}\sum_{j=0}^{N_p} \min\left(I_{DT}(j), K.I'_{DT}(j)\right),$$

with weight values $K \leq 1$ make moving edges more attractive. This edge-based likelihood favours the moving edges without moving the static ones. Consequently, the associated tracker will prefer to stick with the moving edges but in their absence its fallbacks is to use the static ones.

3. Color cue: using discriminant colour (or texture) distributions related to clothes is also of great interest in the weighting stage. By assuming conditional independence of the colour measurements, the likelihood (5) becomes for N_r ROIs

$$p(z^c|x) \propto \exp\left(-\lambda_{p,c} \frac{D^2}{2\sigma_c^2}\right),\ D = \frac{1}{N_r}\sum_{c}\sum_{p=1}^{N_r} D_B(h^c_{x,p}, h^c_{ref,p}), \tag{7}$$

where $\lambda_{p,c}$ are weighting factors discussed in section 5. Each reference histograms $h^c_{ref,p}$ for the $p - th$ patch is learnt on the first sequence image assuming the 3D model has been beforehand superimposed. From (7), we could also define a likelihood $p(z^T_k|x)$ and associated weights $\lambda_{p,t}$ relative to textured patches based on the intensity component.

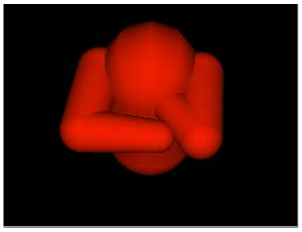


Figure 5. Self-collision example.

4.3 Model priors

1. Non-observable parameters stabilisation: certain pose configuration leads to observation ambiguities as particular DOFs sometimes cannot be inferred from the current image observations. For instance, when one arm is straight and the likelihood $p(z^S|x)$ is used, rotations around the arm's own axis cannot be observed with a coarse, and symmetric body part model. The occluding contour changes little when this limb rotates around its own axis. Leaving this DOF unconstrained would often lead to ambiguities that produce nearly flat cost surfaces. Intuitively, adding texture/colour patches on the arms allows this rotation to be observed. A second and potentially deeper issue introduced in [35] is to control all these hard-to-estimate parameters with a "sticky prior" stabilisers $p_{st}(x) \propto \exp(-\lambda_{st}||x_{def} - x||^2)$ that reaches its minimum on a predefined resting configuration x_{def}. This prior only depends on the structure parameters and the factor λ_{st} will be chosen in a way that the stabilising effect will be negligible for the whole configuration space with the exception of the regions where the visual measurement functions are constant. In the absence of strong observations, the parameters are constrained to lie near their default values x_{def} whereas strong observations unstick the parameters values from these default configurations.

2. Body parts collision detection: physical constraints impose that the body parts do not interpenetrate each other. The admissible volume of the parameter space is smaller than initially designed because certain regions are not physically reachable, and so many false hypothesis may be pruned. Handling such constraints in a continuous optimisation-based framework would be far from trivial. Discrete stochastic sampling framework is clearly more suitable even if some samples may *a priori* fall inside the non admissible parameter space (see an example in Figure 5).

Our strategy is to simply reject these hypothesis thanks to collision detection mechanism. The collision model prior for a state x is thus $p_{coll}(x) \propto \exp(-\lambda_{co} f_{co})$ with $f_{co}(x) = 0$ (resp. 1) whether no collision (resp. in collision). This function, although being discontinuous for some points of the configuration space and constant for all the remaining, is still usable in a Dirac particle filter context. The advantage of its use is twofold, first it avoids the exploration of the filter to zones of no interest, and second it avoids wasting time in performing the measuring step for unacceptable hypothesis as they can be immediately rejected.

Fusing multiple visual cues enables the tracker to better benefit from M distinct measurements $(z^1, \ldots, z^M)$. Assuming that these are mutually independent conditioned on the state, and given L model priors $p_1(x), \ldots, p_L(x)$, the unified measurement function thus factorizes as

$$p(z^1, \ldots, z^M|x) \propto \prod_{i=1}^{M} p(z^i|x). \prod_{j=1}^{L} p_j(x). \tag{8}$$

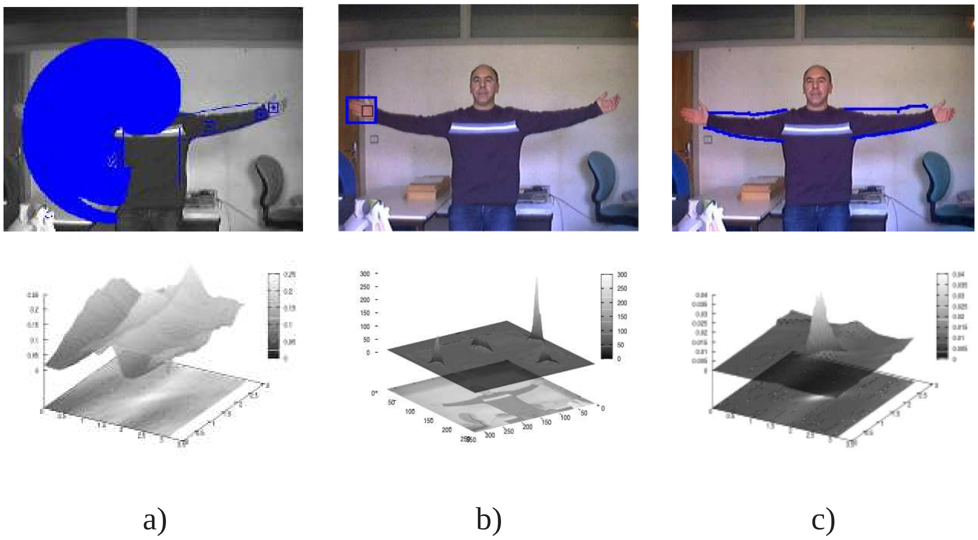
5 Implementation and experiments

In its actual form, the system tracks two-arms gestures under an 8-DOF model, *i.e.* four per arm. We assume therefore that the torso is coarsely fronto-parallel with respect to the camera while the position of the shoulders are deduced from the position of the face given by dedicated tracker [22]. All the DOFs are accounted for in the state vector $\mathbf{x}_k$ related to the k-th frame. Kinematic constraints require that the values of these joint angles evolve within anatomically consistent intervals. Samples (issued from the proposal) falling outside the admissible joint parameter range are enforced to the hard limits and not rejected as this could lead to cascades of rejected moves that slow down the sampler.

Recall that the $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ pdf encodes information about the dynamics of the targeted human limbs. These are described by an Auto-Regressive model with the following form $\mathbf{x}_k^{'} = A\mathbf{x}_{k-1}^{'} + \mathbf{w}_k$ where $\mathbf{x}_k^{'} = [\mathbf{x}_k, \mathbf{x}_{k-1}]^{'}$ and $\mathbf{w}_k$ defines the process noise. In the current implementation, these dynamics correspond to a constant velocity model. We find this AR model gives empirically better results than usual random walk model [33,28].

A set of patches are distributed on the surface model and their possible occlusions are managed during the tracking process. Our approach is different from the traditional marker-based ones because we do not use artificial but natural colour or texture-based markers *e.g.* the two hands and ROIs on the clothes. We adopt the $\mathrm{AUXILIARY}$ scheme (Table 3) which allows to use some low cost measure or *a priori* knowledge to guide the particle placement, therefore concentrating them on the regions of interest of the state space. The measurement strategy is as follows: (1) particles are firstly located in good places of the configuration space according to the initial sampling based on the likelihood (5) in step 6, (2) particles' weights are fine-tuned by combining shape and motion cues, multiple patches per arm as well as model priors thanks to (8) in step 9.

Figure 6-left shows the plot of the shape-based likelihood (6) obtained by sweeping a subspace of the configuration space formed by the orientation of the right arm model involving moderate background clutter. In cluttered background, shape cue is not sufficiently discriminant as multiple peaks are present. Figure 6-middle plots the colour-based likelihood (7) for a single patch corresponding to the marked hand. This is extremely sharp but shows false positive as soon as spurious skin colour like regions are detected. Figure 6-right plots the likelihood $p(z^{MS}, z^{C}|x)$ issued from (8) when fusing shape, motion and colour cues. This plot brings out that this function is more discriminant than the ones involving a single cue.



a) b) c)

Figure 6. Likelihoods for our 3D tracker: a) $p(z_k^S|\mathbf{x}_k)$ contour related likelihood obtained by sweeping the parameter space for 2DoF of the structure; b) $p(z_k^C|\mathbf{x}_k)$ colour related likelihood obtained by sweeping the image with the colour pattern; c) $p(z_k^{MS}, z_k^C|\mathbf{x}_k)$ combined likelihood of the former two showing on the image the projection of the model that corresponds to the peak.

Clearly, mixing all these cues into the measurement function of the underlying estimation scheme helps the tracker to work under a wide range of conditions encountered by our robot during its normal operation.

A second and important line of investigation concerns the incorporation of appropriate degrees of adaptivity into these multiple cues based likelihoods depending on the target appearance and environmental conditions. Therefore, some heuristics allow to weight the strength of each visual cue in the unified likelihood (8). An *a priori* confidence criterion of a given coloured or textured patch relative to clothes can be easily derived from the associated likelihood functions where the p-th colour reference histogram $h^c_{ref,p}$ ($h^t_{ref,p}$ for the texture-related one) is uniform and so given by $h^c_{j,ref} = \frac{1}{N_{bi}},\ j = 1, \ldots, N_{bi}$ where index p has been omitted for compactness reasons. Typically, uniform coloured patches produce low likelihood values, whereas higher likelihood values characterise confident patches because their associated colour distributions are discriminant and ensure non ambiguous matchings. As stated before, parameters $\lambda_{p,c}$ and $\lambda_{p,t}$ weight the strength of the p-th marker in the likelihood (8). In the same way, the parameter λ_s weights the edges' density contribution and is fixed from the first DT image of the sequence.

Due to the efficiency of the importance density and the relatively low dimensionality of the state-space, tracking results are achieved with a reasonably small number of particles *i.e.* $N = 400$ particles. In our unoptimised implementation, a PentiumIV-3GHz runs the two arm tracking process at about $1fps$, being most of the time spent in evaluating the observation function. To compare, classic systems take a few seconds per frame to process a single arm tracking. The fixed parameters involved in the likelihoods, proposal and state dynamics of our upper human body tracker are reported in Table 4.

Symbol	**Meaning**	**Value**
N	number of particles	400
λ_{st}	factor in model prior $p_{st}(x)$	0.5
λ_{co}	factor in model prior $p_{co}(x)$	0.5
K	penalty in likelihood $p(z_k^{MS} \mid \mathbf{x}_k)$	0.5
σ_s	standard deviation in $p(z_k^{MS} \mid \mathbf{x}_k)$	1
N_R	number of patches in colour-based likelihood $p(z_k^C \mid \mathbf{x}_k)$	6
σ_c	standard deviation in $p(z_k^C \mid \mathbf{x}_k)$	0.3
N_{bi}	number of colour bins per channel involved in $p(z_k^C \mid \mathbf{x}_k)$	32

Table 4
Parameter values used in our upper human body tracker.

The above described approach has been implemented and evaluated over monocular image sequences. Figure 7 show an initial test result of tracking a single arm using a 3 degreed of freedom model while figure 8 shows the estimated joint angles values (radians) versus the input frame number.

To illustrate our approach, we show and comment snapshots from typical sequences acquired from the robot Jido (Figure 1) in different situations to highlight our flexible data fusion strategy. The full images as well as other sequences can be found a the URL `www.isr.uc.pt/∼paulo/HRI`. The first sequence (Figure 10) was shot against a white and unevenly illuminated background. Here although using loose fitting clothes, the tracker can follow deictic gestures that demonstrate its ability to work even for poses out of the fronto-parallel plane. The second sequence (Figure 11) involves coarse fronto-parallel motions over a heavy cluttered background. For each snapshot, the right sub-figures show the model configuration for the MMSE estimate, while the left ones show its corresponding estimated configuration corresponding to the posterior pdf $p(\mathbf{x}_k|z_{1:k})$.

The first tracking scenario (Figure 9) shows the estimation of the arms configurations in the presence of a low cluttered background. It can be seen that even with a single camera the system is able to estimate non-planar poses. The second sequence (Figure 9) involves pointing gestures. The target contours are prominent

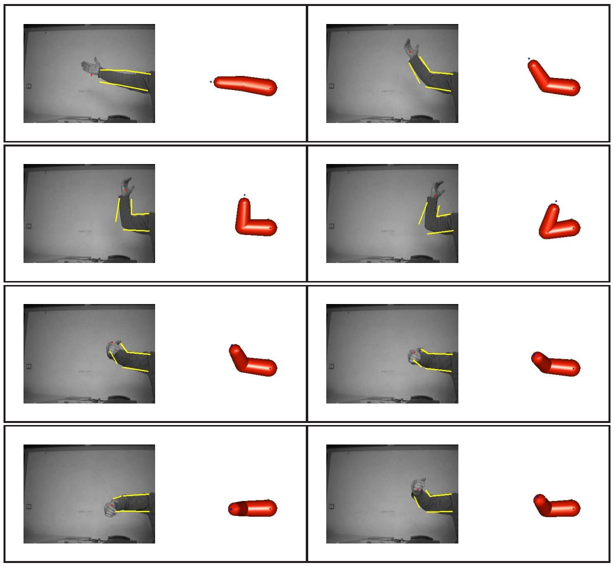


Figure 7. From top-left to bottom-right: snapshots from a simple 3 d.o.f. arm tracking sequence showing the superimposition of the projected model over the input image and the corresponding model configuration

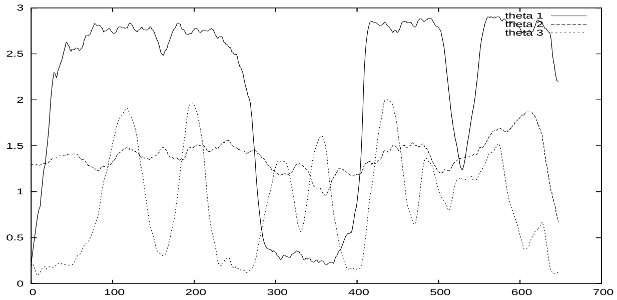


Figure 8. Evolution of the estimated parameters (joint angles) of the model versus the input frame number, for the single arm sequence.

and are weakly disturbed by the background clutter. The high confident contours cue ensure the tracking success. The patches on the uniform sweater are here of little help as their enclosed colour or texture distributions are quite uniform. The flexibility introduced in system by the use of tuneable parameters ($\lambda_{(.)}$), allows to give these patches a weak strength in the unified likelihood cost ($\lambda_{p,c} = 0.1$ against $\lambda_s = 10$). Although they do not introduce any improvement with respect to their position on the arm, they are not completely discarded, as they still contribute with a form of an “inside/outside” information, which complements the contours, specially when they fail. This permitted the tracking of the arms even when they got out of the fronto-parallel plane thanks to all the patches (Figure 10).

For the second scenario (Figure 11), the tracker deals with significantly more complex scene but tracks also the full sequence without failure. This scenario takes clearly benefit from the introduction of discriminant patches as their colour distributions are far from uniform ones. This leads to higher values of confidence ded-

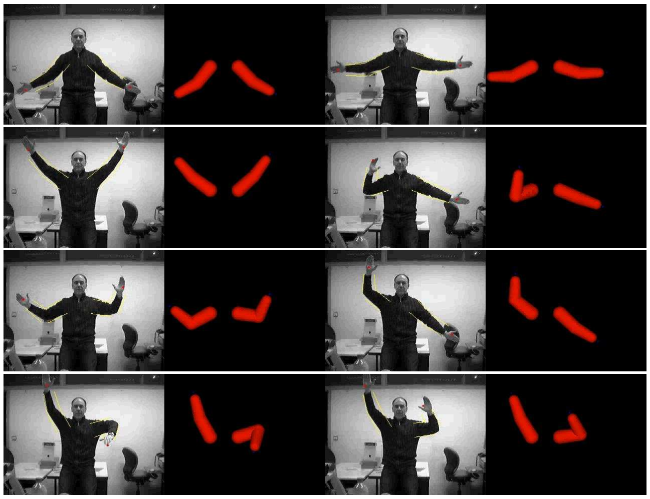


Figure 9. From top-left to bottom right: snapshots from the two arm tracking sequence with a low cluttered background: $\lambda_s = 10, \lambda_{p,c} = 0.1$.

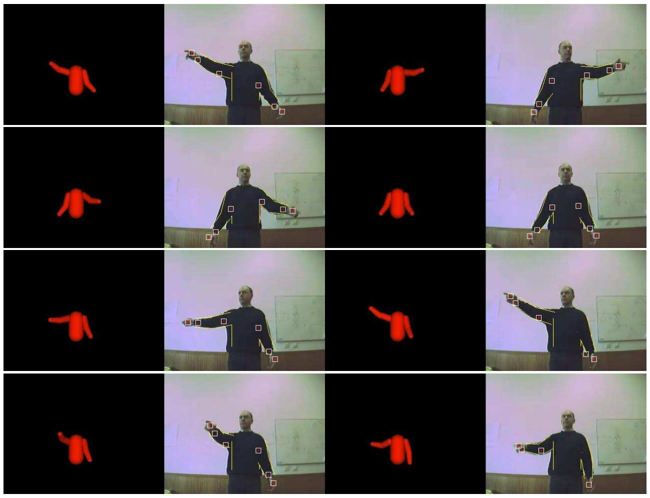


Figure 10. From top-left to bottom right: snapshots of tracking sequence involving deictic gestures: $\lambda_s = 10, \lambda_{p,c} = 0.1$.

icated to the likelihood $p(z_k^C|\mathbf{x}_k)$, namely $\lambda_{p,c} = 1$. In this challenging operating conditions, two heuristics allow jointly to release from distracting clutter that might partly resemble human body parts (for instance the cupboard pillar, whose colour is close to skin colour). On the one hand, estimating the edges density in the first frame highlights that shape cue is not a confident one in this context, so its confidence level in the global cost (8) is reduced accordingly during the tracking process *i.e.* $\lambda_s = 1$. On the other hand, optical flow weights the importance relative to the foreground and background contours thanks to the likelihood $p(z_k^{MS}|\mathbf{x}_k)$. If considering only contour cues in the likelihood, the tracker would attach itself to cluttered background zones and consequently lose the target. Since algorithms correspond to a stochastic framework, we compare multiple runs of the $\mathrm{CONDENSATION}$ and $\mathrm{AUXILIARY}$filters on these two sequences with identical starting state. For a fair empirical comparison, the number of particles used with each filter was chosen

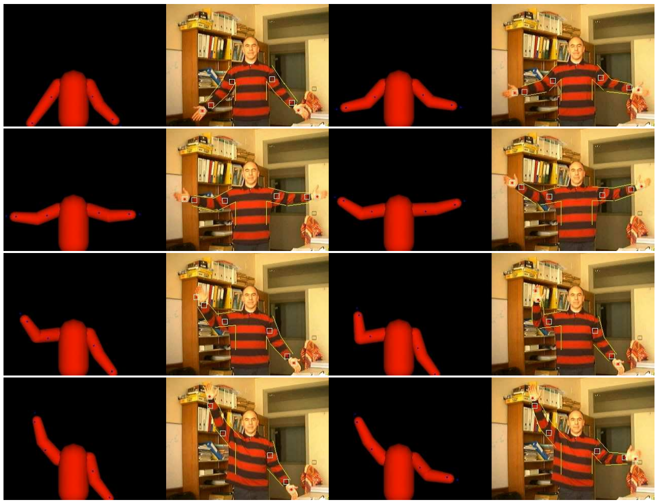


Figure 11. From top-left to bottom-right: snapshots of tracking sequence involving heavy clutter: $\lambda_s = 1, \lambda_{p,c} = 1$.

so that the number of likelihood evaluations per frame was equal. Specifically, 800 particles were used for CONDENSATION and 400 particles for AUXILIARY. The CONDENSATION filter with non flexible data fusion fails in the great majority of runs. In only 10% runs did it provide a correct estimate in terms of the MMSE. The AUXILIARY gave reasonable estimates in 80% runs. Moreover, the AUXILIARY is shown to reduce the variance of the estimate along consecutive trials.

Beyond the aforementioned assistant robot paradigm, another envisaged application concerns the animation of a humanoid robot [23]. This last scenario (Figure 12) with moderate clutter explores 3D estimation behaviour with respect to problematic motions *i.e.* non fronto-parallel ones, elbow end-stops and observation ambiguities. The left column represents the input images and the projection of the model contours superimposed while the right column represents the animation of the HRP2 using the estimated parameters. The first frames involve both elbow end-stops and observation ambiguities. These particular configurations are easily dealt with in our particle filtering framework. When elbow end-stop occurs, the sampler is able to maintain the elbow angle within its predefined hard limits. Observation ambiguity arises when the arm is straight. The twist parameter is temporary unobservable but remains stable thanks to the likelihood $p_{st}(\mathbf{x}_k)$. As highlighted in [6], Kalman filtering is quite unable to track such particular arm configurations. Some frames later in figure 12, the left arm bends slightly towards the camera. Thanks to the patches on the hands, the tracker manages to follow this temporary unobservable motion, although it significantly mis-estimates the rotation during this motion. The entire HRP2 video is available at `www.isr.uc.pt/∼paulo/HRI`.

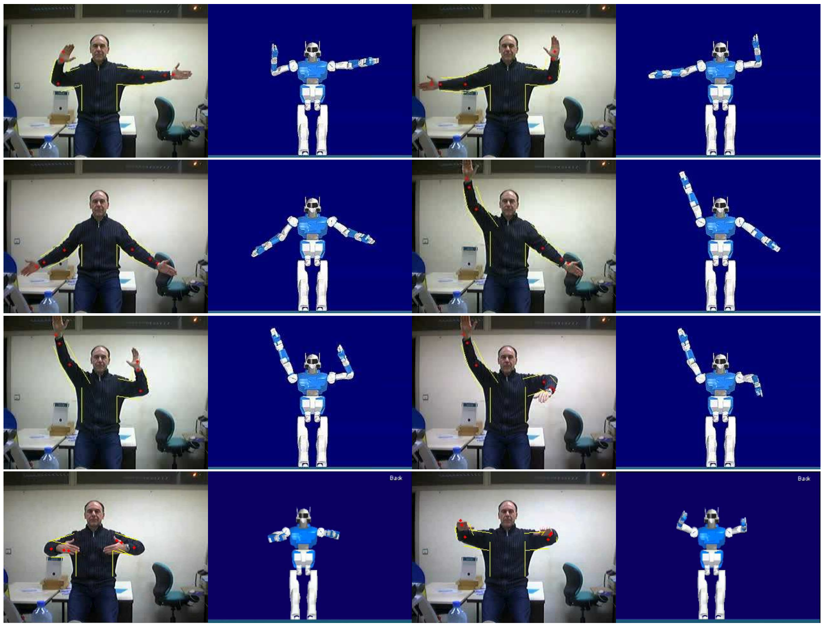


Figure 12. From top-left to bottom-right: snapshots of tracking sequence and animation of HRP2 using the estimated parameters: $\lambda_s = 5, \lambda_{p,c} = 0.5$.

6 Conclusion

This article presents an innovative particle filtering framework for 3D tracking the upper human body parts using a single camera mounted on an assistant mobile robot. Like numerous approaches, the principle relies on the model image projection and model-image matching cost metric to infer the model configuration in 3D. Our particle filter based HMC system differs from conventional particle filters as follows. The genuine AUXILIARY strategy limits drastically the well-known burst in terms of particles when considering high dimensional state-space. This nice property is due to both dynamic and image data driven proposal density. Data fusion is also considered in the measurement function. The weighting stage relies on a new model-image matching cost metric, which combines, in a flexible way, extracted edges (weighted by flow), coloured (or textured)- based patches, kinematic joint limits, and non self-intersection constraints. Experiments on monocular sequences acquired from the robot, show that the proposed framework is suitable for tracking human motions, and that the flexible integration of multiple cues improve the tracker versatility. Moreover, our tracker is applied in a quasi-real-time process and so requires less computational power than most of the existing approaches. Combined with today's powerful off-the-shelf PCs, such quasi real-time HMC approach devoted to mobile robot nearly becomes a reality and would have a great number of robotic applications.

Several directions are studied regarding our visual modalities. Firstly, we are currently working on the idea of continuously learn the appearance of new patches, distributed all over the surface of the model, during the tracking process. Next, our measurement function will be enriched with sparse 3D reconstruction-based data. Sparse 3D-point cloud will aid recovery from transient tracking failures and will free the tracker from the classical *ad hoc* initialisation. Further evaluations will be also performed using a motion capture testbed that provides more accurate ”ground truth” from a commercial HMC system, such as VICON, that will be synchronised with the video streams.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *Trans. on Signal Processing*, 2(50):174–188, 2002.

[2] P. Azad, A. Ude, T. Asfour, and R. Dillmann. Stereo-based markerless human motion capture for humanoid robot systems. In *Int. Conf. on Robotics and Automation (ICRA’07)*, pages 3951–3956, Roma, Italy, April 2007.

[3] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with physical forces. *Int. Journal Computer Vision and Image Understanding (CVIU’01)*, 81:328–357, 2001.

[4] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR’00)*, pages 126–133, Hilton Head, USA, June 2000.

[5] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR’01)*, pages 669–676, 2001.

[6] J. Deutscher, B. North, B. Bascle, and A. Blake. Tracking trough singularities and discontinuities by random sampling. In *Int. Conf. on Computer Vision (ICCV’99)*, Corfu, Greece, September 1999.

[7] A. Doucet, N. De Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Series Statistics For Engineering and Information Science. Springer-Verlag, New York, 2001.

[8] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[9] J. Gall, B. Rosenhahn, and H.-P. Seidel. An introduction to interacting simulated annealing. In Klette R., Metaxas D., and Rosenhahn B., editors, *Human Motion - Understanding, Modeling, Capture and Animation*, volume 36 of *Computational Imaging and Vision*, pages 319–345. Springer-Verlag, 2008.

[10] D.M. Gavrila. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. Journal of Computer Vision (IJCV’07)*, 73(1):41–59, 2007.

[11] D. Geronimo, A.M. Lopez, A. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *Trans. on Pattern Analysis Machine Intelligence (PAMI'10)*, 28(6):976–990, 2010.

[12] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Janson, R. Karlsson, and P-J Nordlund. Particle filters for positioning, navigation and tracking. *Trans. on Signal Processing*, 50(2):425–437, 2002.

[13] N. Hasler, B. Rosenhahn, Thormahlen T, W. Wand, J. Gall, and H.-P. Seide. Markerless motion capture with unsynchronized moving cameras. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'09)*, Miami, USA, June 2009.

[14] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[15] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conf. on Computer Vision (ECCV'96)*, pages 343–356, Cambridge, UK, April 1996.

[16] M. Isard and A. Blake. I-CONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *European Conf. on Computer Vision (ECCV'98)*, pages 893–908, Freiburg, Germany, June 1998.

[17] M. Jones and J. Rehg. Color detection. Technical report, Compaq Cambridge Research Lab, 11 1998.

[18] J. Kenna and H. Nait-Charif. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Int. Journal Image and Vision Computing (IVC'07)*, 25(6):852–862, 2007.

[19] S. Knoop, S. Vacek, and R. Dilman. Sensor fusion for 3D human body tracking with an articulated 3D body model. In *Int. Conf. on Robotics and Automation (ICRA'06)*, pages 1686–1691, Orlando, USA, May 2006.

[20] M.W. Lee, I. Cohen, and S.K. Jung. Particle filter with analytical inference for human body tracking. In *Workshop on Motion and Video Computing (MOTION'02)*, Orlando, USA, December 2002.

[21] B. Leibe, K. Schindler, N. Cornelis, and L.V. Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Int. Journal Trans. on Pattern Analysis and Machine Intelligence (PAMI'08)*, 30(10):1683–1698, 2008.

[22] P. Menezes, J.C. Barreto, and J. Dias. Face tracking based on haar-like features and eigenfaces. In *IFAC Symp. on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[23] P. Menezes, F. Lerasle, J. Dias, and R. Chatila. A single camera motion capture system dedicated to gestures imitation. In *Int. Conf. on Humanoid Robots (HUMANOID'05)*, pages 430–435, Tsukuba, Japan, 2005.

[24] T. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Int. Journal Computer Vision and Image Understanding (CVIU'06)*, (104):90–126, 2006.

[25] T.M. Mutali. *Efficient Hidden-Surface Removal in Theory and in Practice*. PhD thesis, Dept. of Computer Science, Brown University, 1998.

[26] J. Nierverget and F.P. Preparata. Plane sweeping algorithms for intersecting geometrical figures. *Communications of ACM*, 25:739–747, 1982.

[27] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptative color-based particle filter. *Int. Journal Image and Vision Computing (IVC'03)*, 21(90):90–110, 2003.

[28] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Int. Journal IEEE*, 92(3):495–513, 2004.

[29] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446), 1999.

[30] R. Pope. Vision-based human motion analysis: an overview. *Int. Journal Computer Vision and Image Understanding (CVIU'07)*, (108):4–18, 2007.

[31] D. Ramanan, D.A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, USA, June 2004.

[32] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, pages 786–793, Hawai, December 2001.

[33] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *European Conf. on Computer Vision (ECCV'00)*, pages 702–718, Dublin, Ireland, July 2000.

[34] L. Sigal, S. Bhatia, S. Roth, J. Black, and M. Isard. Tracking loose-limbed people. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, USA, June 2004.

[35] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *Int. Journal on Robotic Research (IJRR'03)*, 6(22):371–393, 2003.

[36] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *British Machine Vision Conf. (BMVC'01)*, volume 1, pages 63–72, Manchester, UK, September 2001.

[37] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Int. Conf. on Computer Vision (ICCV'03)*, pages 1063–1070, Nice, France, October 2003.

[38] I.E. Sutherland, R.F. Sproul, and R.A. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Computing Survey*, 1(6):1–55, 3 1974.

[39] P. Torma and C. Szepesvari. Sequential importance sampling for visual tracking reconsidered. In *AI and Statistics*, pages 198–205, 2003.

[40] R. Urtasun and P. Fua. 3D human body tracking using deterministic temporal motion models. In *European Conf. on Computer Vision (ECCV'04)*, Prague, Czech Republic, May 2004.

[41] X. Xu and B. Li. Learning motion correlation for tracking articulated human body with a rao-blackwellised particle filter. In *Int. Conf. on Computer Vision (ICCV'07)*, Rio de Janeiro, Brazil, October 2007.

[42] J. Ziegler, K. Nickel, and R. Stiefehagen. Tracking of the articulated upper body on multi-view stereo image sequences. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, New York, USA, June 2006.