Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots

# A Single Camera Motion Capture System dedicated to Gestures Imitation[*]

Paulo Menezes[*][†], Frédéric Lerasle[*], Jorge Dias[†], Raja Chatila[*]
[*]Laboratoire d'Architecture et Analyse des Systèmes - LAAS-CNRS
7 av du colonel Roche, 31077 Toulouse cedex 4, France
Email: {pmenezes,lerasle,raja}@laas.fr
[†]Instituto de Sistemas e Robótica, University of Coimbra
Polo II - 3030-290 COIMBRA, Portugal
Email: {paulo,jorge}@isr.uc.pt

*Abstract*— **This article describes a method to track 3D articulated structures from a monocular perspective image sequence using a particle filter. This structure is composed of a set of linked degenerated quadrics (cones) which are truncated by pairs of planes also modelled as degenerated quadrics. This set of truncated cones is connected by joints containing one or more degrees of freedom. The method is based upon the estimation of the contours of the structure's silhouette projected on the image plane, to validate each of the particles which correspond to proposed configurations. This validation is performed using a criterion that combines a measure based on the contours, a measure of similarity between the colour distribution around a point on the structure and a reference distribution, and other criteria that reinforce the overall cost function. The results show the feasibility of the approach when using a single camera to track a 3D model containing eight degrees of freedom.**

*Index Terms*— **3D tracking, monocular vision, articulated structures, particle filters, gesture imitation**

## I. Introduction

In the recent years, a great effort in the Robotic community was devoted to the realization of a humanoid robot which can imitate human motion e.g. the dancing humanoid robot proposed by Ikeuchi et al. [1]. The standard approach consists in analyzing raw motion data acquired from the motion capturing system VICON. This system allows to acquire time series positions of markers placed at special anatomical points of the tracked subject. The markers are passive (reflector markers) or active (infrared blankers, electro-luminescent light emitting diodes). The active methods make the recognition of the markers easier because each LED emission can be analyzed separately.

Using such motion capturing systems is not intuitive and questionnable in this context. Firstly, captured motion cannot be directly imported into a robot, as the raw data must be converted to its joint angle trajectories. The trajectories are here modified to be feasible ones within the constraints of this robot. Our motion capture system outputs data indeed coarser but more suited to a humanoid robot. Secondly, usual motion capture systems are hard to implement while using markers is restrictive. We aim to investigate marker-free motion capturing system considering a single camera.

To date, most of the existing markerless approaches [2]–[5] use a prior 3D model of the viewed human limbs, together with either 3D reconstruction approaches or appearance based approaches, being the latter chosen for this work. This kind of approach is based on the generation of the expected projection of parts of the model, which are then compared to equivalent features extracted from the input images.

Actually, the complexity of the models depends on the required precision and on the involved algorithms. In our case, building a simple and light approach that would be adequate to be applied in a quasi-real-time application was one of the ideas that guided the developments since the begining. This motivated our choice of using quadrics for constructing the models, because their handling is simple, they can be combined to create complex shapes and, their projections are conic sections that can be obtained in closed form.

Most of the classical approaches to the monocular tracking problem are normally limited to movements that are roughly contained in a fronto-parallel plane. Our approach, enlarges the spectrum of the tracked movements by exploring the model appearance and the kinematic constraints of the structure, still using a single camera.

The noisy measures, the relative complexity of the models, and their shapes (revolution shaped) give frequently rise to singularities in the state-measure link. To solve these ambiguities some authors insist on the interest of considering multiple hypotheses at each time step for the parameters to be estimated [6], [7]. We present an appearance based tracking method that combines some expected appearance attributes of the 3D model to validate the hypotheses proposed by a particle filter.

The particle filter framework [8], [9], has two important advantages: it is not dependent of any underlying probabilistic model (e.g. Gaussian) that may characterise the distributions, and it can easily use combined information from different sources of measure in the particle weighting step.

This article is organised as follows. Section II focuses on the modelling of the 3D structure and section  III presents our strategy to generate its projection and handle any concealing that may occur between parts. The application to visual tracking is depicted in section IV. The implementation and the

[*]This work is partially supported by an FCT Grant to P. Menezes

0-7803-9320-1/05/$20.00 ©2005 IEEE

obtained results are presented on section V. The ouput motion data devoted to the HRP2 humanoid for gestures imitation shows the reliability of our approach. Section VI concludes the article and proposes some future extensions to this work.

## II. Modelisation of the human limbs

An approximative model was build where the human limbs are represented by 3D articulated structures composed of truncated quadrics. These geometric primitives can be easily manipulated and projective geometry provides the tools to obtain their projections in an elegant way. The truncated quadrics are connected between them by articulations that represent the corresponding human articulations, e.g. elbow, shoulder, etc. Depending on the case each articulation can contain one or more degrees of freedom.

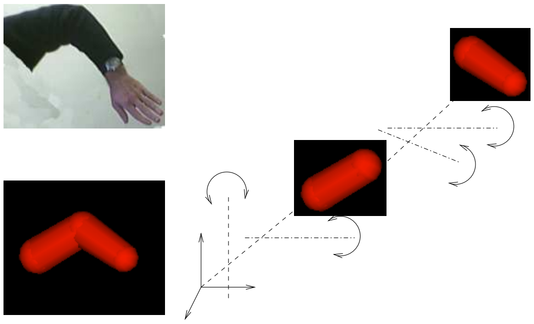

Fig. 1. Arm structure exhibiting its degrees of freedom.

Figure 1 represents the model for an human arm containing four degrees of freedom, where both the forearm and the upper arm are modelled by truncated cones.

As in [10] and [4], the parts that compose the arm model form a kinematic chain. Thus, if a transformation is applied to part $i$, as a consequence of the positioning of one of its degrees of freedom, all the subsequent parts $(i+1)$ in the chain (or chains) will undergo through the same transformation.

### A. Modelling the structure with quadrics

A quadratic surface, or quadric, is an implicit surface of second order. It can be represented in vectorial form using homogeneous coordinates as

$$\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0 \tag{1}$$

where $\mathbf{x} = \begin{bmatrix} x & y & z & w \end{bmatrix}^T$ and $\mathbf{Q}$ is a $4\times 4$ symmetric matrix. Depending on the properties of the $\mathbf{Q}$ matrix, the expression (1) can represent 17 different types of quadrics. However, only three of these are of interest for this work, which are: ellipsoids, cones and cylinders, and pairs of planes.

These quadrics can be combined in different ways to build more complex forms. As an example, a truncated cone can be defined by the set of points that verify:

$$\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0 \quad \wedge \quad \mathbf{X}^T\Pi\mathbf{X} \leq 0. \tag{2}$$

where $\mathbf{Q}$ and $\Pi$ are the matrices that correspond to the cone and to the pair of planes that delimits the cone, respectively.

Finally, it should be noted that a quadric $Q^{'}$ obtained through the application of the transformation $\mathbf{T}$ to the points of the surface $\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0$ is $\mathbf{X}^{'T}\mathbf{Q}^{'}\mathbf{X}^{'} = 0$ with $\mathbf{X}^{'} = \mathbf{T}\mathbf{X}$ and $\mathbf{Q}^{'} = \mathbf{T}^{-T}\mathbf{Q}\mathbf{T}^{-1}$.

## III. Generating the appearance model

This section explains the method we developed to obtain the projection of the above defined model, considering the possibility that a part can hide another one, either completely or partially.

Lets start with the projection of a quadric in a normalised camera, i.e. one whose projection matrix is $\mathbf{P} = [\mathbb{I}_{3\times 2}|\mathbf{0}_{3\times 1}]$.

For a pinhole camera model, a projection ray tangent to the surface of the quadric intersects the image plane at a point, $\mathbf{p} = [x\ y\ 1]^T$, shown on figure 2. Then, every point on this projection line can be defined by $\mathbf{X} = \begin{bmatrix} \mathbf{x}^T & s \end{bmatrix}^T$ where $s$ is a scalar. By making $\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix}$, and substituting into equation (1) we get

$$\begin{bmatrix} \mathbf{x} \\ s \end{bmatrix}^T \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & c \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ s \end{bmatrix} = 0 \tag{3}$$

or

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + 2s\mathbf{b}^T\mathbf{x} + s^2c = 0. \tag{4}$$

Although this equation is quadratic in $s$, there is a single solution for any image point which corresponds to the projection of a tangency point of the model. Thus, the set of points that belong to the silhouette contours must verify the following expression:

$$\mathbf{x}^T(\mathbf{b}\mathbf{b}^T - cA)\mathbf{x} = 0. \tag{5}$$

Now it is possible to infer each of the 3D points by obtaining the corresponding $s$ value, which is given by

$$s = -\mathbf{b}^T\mathbf{x}/c \tag{6}$$

This method can be extended to the general camera case, $\mathbf{P}'$, by defining a matrix $H_{4\times 4}$ such that $\mathbf{P'H} = \begin{bmatrix} \mathbb{I} & \mathbf{0} \end{bmatrix}$, and then performing a small manipulation gives:

$$\begin{aligned} \mathbf{x} &= \mathbf{P'X} \\ &= \begin{bmatrix} \mathbb{I} & \mathbf{0} \end{bmatrix}\mathbf{H}^{-1}\mathbf{X}. \end{aligned} \tag{7}$$

This shows that the projection of a point $\mathbf{X}$ under the hypothesis of a real camera is equivalent to the projection of the point $\mathbf{X}^{'} = \mathbf{H}^{-1}\mathbf{X}$ under the hypothesis of a normalised camera.

It is also easy to show that the projection of the quadric of matrix $\mathbf{Q}$ under a real camera is equivalent to the projection of the quadric represented by $\mathbf{Q}^{'} = \mathbf{H}^T\mathbf{Q}\mathbf{H}$ using a normalised camera.

### A. Projection of the models' geometrical components

The image of a degenerated quadric, obtained by projective projection, is a degenerated conic. 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 recognise if the matrix $\mathbf{C_p} = \mathbf{b}\mathbf{b}^T - c\mathbf{A}$ is diagonal.

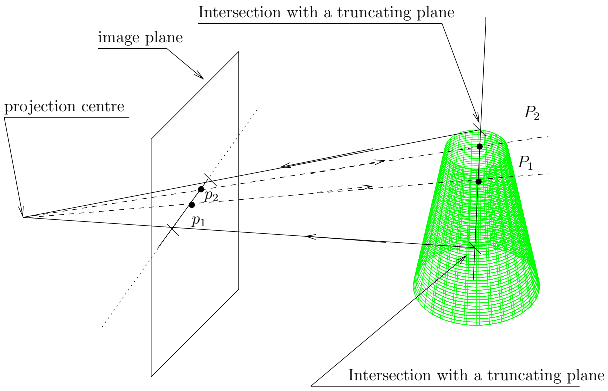
Intersection with a truncating plane
image plane
projection centre
$P_2$
$P_1$
$p_2$
$p_1$
Intersection with a truncating plane

Fig. 2. Truncating projected segments

This corresponds, in fact, to the case where the conic is centred at the origin and its axes are aligned with the ones of the image plane. It suffices then to estimate a transformation $\mathbf{B}_{3\times3}$ that, once applied, will bring the conic into this particular configuration and thence diagonalising the matrix $\mathbf{C_p}$. Now for the more general cases where there are two lines, we can easily determine two points on each of them that we denote $\mathbf{p}_1$ and $\mathbf{p}_2$. These points can be brought back to their original configuration through the transformation $\mathbf{B}^{-1}$, defining each of the projected lines.

To truncate the cone projection, and thus obtain the required line segments, the 3D points $\mathbf{P}_i$ associated with the image points $\mathbf{p}_i$ are computed by

$$\mathbf{P}_i = \left[ \begin{array}{c} \mathbf{p_i} \\ -\frac{\mathbf{b}^T\mathbf{p_i}}{c} \end{array} \right] \tag{8}$$

where $\mathbf{b}$ and $c$ are blocks of $\mathbf{Q}$. The visibility frontier of the model (that one that projects over the silhouette contour) passes on the two points $\mathbf{P}_1$ and $\mathbf{P}_2$, so, every 3D point on this line verifies:

$$\mathbf{P}_n = \mathbf{P}_1 + \lambda\mathbf{P}_2. \tag{9}$$

The intersection between this line and the two clipping planes is given by

$$(\mathbf{P}_1 + \lambda\mathbf{P}_2)^T \Pi (\mathbf{P}_1 + \lambda\mathbf{P}_2) = 0. \tag{10}$$

Solving this equation gives two values for $\lambda$ that, once substituted into expression (10), produce the required 3D (intersection) points. These ones, are reprojected 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.

### B. Hidden parts handling

For a given configuration and/or pose, parts of the articulated model can conceal other ones either in a partial or complete way. Being the result of projecting the model a set of line segments, it is necessary to remove the hidden segments or their hidden parts.

There are several algorithms available in the literature to manage the hidden parts of a projected model ( [11], [12]). Most of them are computationally heavy or inadequate for the current problem. A recent algorithm presented in [4], although being quite adequate to the problem, presents a complexity which depends 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 algorithm whose complexity depends only on the number of projected parts and not on their size or precision. This algorithm requires, consequently, less computational power than the former ones.

The first step consists on the computation of all the strict intersections amongst the whole set of projected segments. Strict intersection between two segments is 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 [13], 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.

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 have any strict intersection between them.

Then, the middle point, $p_m$, of each segment, is used to define a projective line passing through $O$ (the camera centre) and $p_m$. The intersections between this line and each quadric that compose the model (see figure 3) allow to conclude about the visibility of the segment. It should be said that a single test on the segment's middle point is enough to infer about the visibility of a segment.

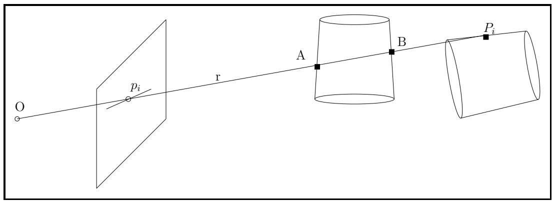

Fig. 3. Visibility test for projected the segments

## IV. Estimating the structure parameters from the video stream

The goal is to estimate the configuration parameters of the 3D structure that generate a projection that best fits to each of the images of the sequence representing the structure. This estimation problem is ill-conditioned, as some d.o.f.'s are not continuously observable from a single camera. Although there is no robust method to infer the structure parameters directly from the observations, it is still possible to build a cost function that reaches its minimum for the true configuration. We propose an approach that incorporates kinematic and geometric constraints, as well as different sources of measure in the construction of such cost function.

The particle filter method approximates the probability distribution of the state as a sum of Dirac impulses, which evolve throughout the prediction, measurement and update steps of the algorithm.

The set of parameters to be estimated is collected in the state vector

$$\mathbf{x}_k = [\ \theta_{1,k} \quad \theta_{2,k} \quad \cdots \quad \theta_{n,k}\ ]^T$$

and the dynamics are described by an auto-regressive model of the following form

$$\mathbf{x}'_k = \mathbf{A}\mathbf{x}'_{k-1} + \mathbf{w}_k \tag{11}$$

where $\mathbf{x}'_k = \left[\ \mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T\ \right]$. The index $k$ is related with the $k$-th image of the sequence and $\mathbf{w}_k$ defines the process noise. This dynamics model is used in the prediction step of the filter, and the predicted hypotheses about the structure configuration are then validated using a measurement step that performs a confrontation between these predicted configurations and the input image. So, the measurement model should enable to weight more or less heavily a particle whether the associated model projection is more or less coherent with the observed input image. This measurement model is described in the next subsections.

### A. Image contours

The first term of the cost function is based on the matching level between the predicted projected contours and the extracted image edges.The principle is to create a function based on the distance between a set of sampling points along the projected contours, for a given model configuration, and the closest ones of the set of image edges. The lower this cost function is, the better is the matching between the image and the model projection.

This measure can be constructed from a distance transform (DT) image, where the low valued points are situated in the neighbourhood of an edge and the high valued ones are away from the edges.

The weight, $k_k^i$, of the particle $i$ at time $k$, $\mathbf{x}_k^i$, must relate the configuration this particle proposes and the input image using the edge and contour information. So a distance like function can be built by uniformly sampling the distance transform along the visible projected line segments. This gives the following distance measure

$$d_{ch}^i = K_{ch} \sum_{j=0}^{N} d^i(j). \tag{12}$$

where $d^i(j)$ refers to the value of the distance transform at sampling point number $j$, on the projected segments, with $j = 1, \ldots, N$.

### B. Colour distribution

Some parts of the model may present discriminative colour distributions that will easily distinguished from other areas of the image.

The colour distribution on the surface of an object is frequently an important characteristic of the tracked targets that can be used in some object tracking applications. In the current case, it was verified that the colour distribution over the hand surface is an important cue that can be used to constrain the position of the forearm and, consequently, the whole model.

To represent the colour distribution, independent normalised histograms in the RGB space were used. Being $h_{ref}^c$ the histogram of reference with $N_b$ bins connected to the skin colours, the colour channel $c$ can be indexed, such ($c \in \{R, G, B\}$):

$$h_{ref}^c = (h_{1,ref}^c, \ldots, h_{N_b,ref}^c)$$

The colour distribution $h_x^c = (h_{1,x}^c, \ldots, h_{B,x}^c)$ of a region of interest $B_x$ that corresponds to the system state $x$ is given by:

$$h_{j,x}^c = c_H \sum_{u \in B_x} \delta_j(b_u^c), j = 1, \ldots, N_b$$

where $b_u^c \in \{1, \ldots, N_b\}$ indexes the histogram's bin that corresponds to the level of pixel $u$ for channel $c$, $\delta_a$ is the Kronecker symbol at $a$, and $c_H$ is a normalisation term such that:

$$\sum_{j=1}^{N_b} h_{j,x}^c = 1.$$

From within the similitude measures that are available to compare two distributions $h_1 = \{h_{j,1}\}_{j=1,\ldots,N_b}$ and $h_2 = \{h_{j,2}\}_{j=1,\ldots,N_b}$, our choice felt on the Bhattacharyya coefficient

$$\rho(h_1, h_2) = \sum_{j=1}^{N_b} \sqrt{h_{j,1}.h_{j,2}}$$

due to its simplicity. A distance function was then defined as

$$d^i(h_1, h_2) = (1 - \rho(h_1, h_2))^{1/2}$$

which can be applied to the three RBG channels, giving

$${d_{bh}^i}^2 = K_{bh} \sum_{c \in \{R,G,B\}} d^{i^2}(h_x^c, h_{ref}^c)$$

### C. Optical Flow

The scene background clutter may contain many edges that, when compared to the projected contours, produce local minima in the contour-edge cost function. To deal with this kind of problems we consider an assumption, that is a realistic one for some scenarios, and which is: the tracked structure is a moving over a static background. Moving regions induce in the image plane a field of normal optical flow and the cost functions based on contour comparison can be scaled locally using a optical flow based function. The distance transforms can be computed for both filtered and non-filtered contour images and using relation (12) we can obtain

$$d_f^i = \sum_{j=0}^{N} \min\left(d_{all}^i(j), K_{of}.d_{of}^i(j)\right). \tag{13}$$

where $K_{of}$ is a constant, $d_{all}^i(j)$ is the value of the DT for the j-th point of the projected contours, and $d_{of}^i(j)$ is the value of the DT for the j-th point of the projected contours filtered by the optical flow.

### D. Non-observable parts stabilisation

Frequently parts of the structure are not observable and thus no reasoning can be made about their parameters. This introduces an ambiguity in the estimation problem if these parameters are left unconstrained. Like in [14], a cost function can be built, that reaches its minimum on a predefined resting configuration $\mathbf{x}_{def}$ (or on the last one estimated). This enables the saving of computing efforts that would explore the unobservable regions of the configuration space and, instead of that, will focus on the estimation of observable parameters. This will ensure also, that in the absence of strong measurements, the parameters will be stabilised. As more constraints from measurements become available, they may move the parameters values from these default configurations. This can be obtained by using a weak cost function that reaches its minimum about the default configuration, which can be defined as

$$f_s = k_s ||\mathbf{x}_{def} - \mathbf{x}_k^i||^2. \tag{14}$$

This function only depends on the structure parameters and the factor $k_s$ will be chosen in a way that the effect of $f_s$ will be negligible most of the time with the exception of the regions of the configuration space where the other cost terms are constant.

### E. Collision detection

For kinematic redundant structures, some combination of parameters, although within the physical limits of the joints, may correspond to collisions or inter-penetration of parts of these structures. In the particle filter framework it is possible that configurations proposed by some particles correspond to such situations, thus wasting computing power exploring regions of the configuration space that are of no interest.

To avoid these situations, a binary cost function, that is not related to observations, was proposed based on a collision detection mechanism. This function is of the form:

$$f_c(\mathbf{x}_k^i) = \begin{cases} 0 & \text{No collision} \\ 1 & \text{In collision} \end{cases} \tag{15}$$

Such cost function, although being discontinuous for some points of the configuration space and consequently cannot be linearised around them, and being constant for all the remaining, is still usable in a Dirac particle filter context.

### F. Combined measurement model

The weighting operation of each particle $x_k^i$ is performed by a weighted combination of the previously defined cost functions, whose contributions are balanced by the set of parameters $(\lambda_1, \lambda_2, \lambda_3)$. The final weighting function is by consequence:

$$w_k^i = \exp - \left( d_f^i(\mathbf{x}_k^i) + \lambda_1 . d_{bh}^i(\mathbf{x}_k^i) + \lambda_2 . f_s(\mathbf{x}_k^i) + \lambda_3 . f_c(\mathbf{x}_k^i) \right) \tag{16}$$

## V. Implementation and experimental results

The above described methods have been implemented and tested over monocular sequences of images representing the motion of two human arms. Contrary to other works [2], [4] that use multi-ocular approaches to track motions not constrained to fronto-parallel planes, this approach is monocular. This was validated by tracking simultaneously two human arms using an 8 d.o.f. Figure 4 shows on the left column the input images with the projection of the model contours superimposed and on the right column the animation of the HRP2 using the estimated parameters. The evolution of these estimated parameters (angles versus the frame numbers) is plotted in figure 5. The weighting function integrates not only contour and colour information, but also collision detection and optical flow based contour filtering. The presence of background clutter as well as the increase on the dimensionality the problem required the introduction of the two last terms in the cost function. The goal of these additional terms is twofold: (1) to make the tracker less disturbed by background clutter, and (2) to reduce the influence of false local minima so that a small number of particles be still enough to solve the minimisation problem. The hidden part stabilisation term was not included in these first experiments because mutual hiding between parts did not occur. The input sequences were previously saved using a "webcam" at a frame rate of 15Hz. This implementations of the algorithm ran on a PentiumIV-3GHz machine at a rate of 2 Hz, although no special care was taken with respect to code optimisation.

The tracker initialisation was performed manually for the first image of the sequence, by choosing the initial joint angles and selecting an image region to generate the reference colour histogram of the hand. These results were obtained using a filter composed of 500 particles, whose number was chosen heuristically. The video sequence here presented is available at the address `http://www.laas.fr/~pmenezes/Humanoids05/`.

## VI. Conclusion

This article presents a method for performing human motion capture using a single camera and an approximative articulated model. It relies on a particle filtering technique to estimate the various parameters of the structure that are related with the kinematic degrees of freedom. The tracking principle is based on the comparison of the expected appearance of the model to the input image.

The model is constructed using degenerated quadrics as they provide a good approximation of the limbs (for the current application) and an elegant way of generating the corresponding perspective projection. Based on this kind of model a strategy was proposed to handle the projection and hidden parts removal efficiently. Experiments showed that our approach could be used to animate an humanoid robot like HRP2.

Future works will be devoted to cases where the model contains a higher number of d.o.f. Until now we have used

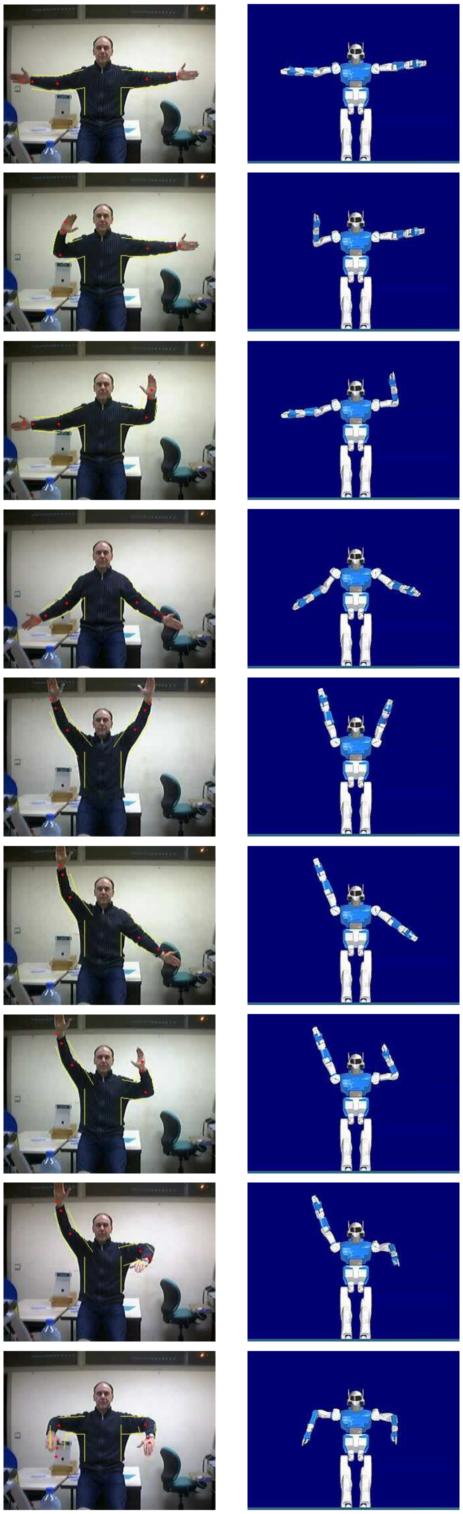

Fig. 4. Left: superimposition of the projected model over input image; right: animation of HRP2 using the estimated parameters

a classical particle filter algorithm, but we are convinced that exploring some variants available in the literature it will be possible to keep the number of particles within acceptable limits for more complicates structures.

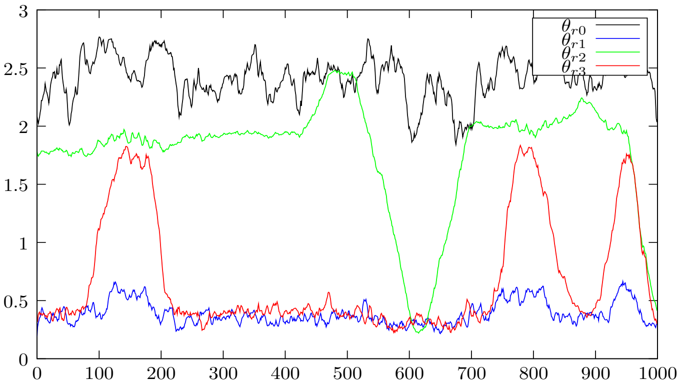

Fig. 5. Evolution of the estimated parameters (joint angles in radians) for the right arm of the model versus the input frame number.

## ACKNOWLEDGMENTS

This animation was performed using the KineoWorks™ platform and the HRP2™ model by courtesy of AIST (GeneralRobotix).

The authors would like also to thank C. Esteves for her help in interfacing with the KineoWorks™ platform and models.

## REFERENCES

[1] A. Nakazawa, S. Nakaoka, S. Kudo, and K. Ikeuchi, “Imitating human dance motion through motion structure structure analysis,” in *Int. Conf. On Robotics and Automation*, 2002.

[2] Q. Delamarre and O. Faugeras, “3D articulated models and multi-view tracking with silhouettes,” in *Int. Conf. Computer Vision*, 199, pp. 716–721.

[3] K. Rohr, “Towards model-based recognition of human movements in image sequences,” in *CVGIP: Image Understanding*, vol. 1, no. 59, 1994, pp. 94–115.

[4] B. Stenger, P. R. S. Mendonça, and R. Cipolla, “Model-based hand tracking using an unscented kalman filter,” in *Proc. British Machine Vision Conference*, vol. 1, Manchester, UK, 9 2001, pp. 63–72.

[5] 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 (2)*, 2000, pp. 702–718.

[6] Y. Wu, J. Y. Lin, and T. S. Huang, “Capturing natural hand articulation,” in *Int. Conf. On Computer Vision*, 2001, pp. 426–432.

[7] J. Deutscher, A. Blake, and I. Reid, “Articulated body motion capture by annealed particle filtering,” in *Int. Conf. On Computer Vision and Pattern Recognition*, 2000, pp. 126–133.

[8] A. Doucet, N. de Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*. Springer-Verlag, 2001, pp. 3–14.

[9] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking,” *IEEE Trans. On Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[10] J. M. Rehg and T. Kanade, “Visual tracking of high dof articulated structures: an application to human hand tracking,” in *European Conference On Computer Vision (2)*, 1994, pp. 35–46.

[11] T. M. Mutali, “Efficient hidden-surface removal in theory and in practice,” Ph.D. dissertation, Dept. of Computer Science, Brown University, 1998.

[12] I. E. Sutherland, R. F. Sproul, and R. A. Schumacker, “A characterization of ten hidden-surface algorithms,” *ACM Computing Survey*, vol. 1, pp. 1–55, 3 1974.

[13] J. Nierverget and F. P. Preparata, “Plane sweeping algorithms for intersecting geometrical figures,” *Communications of ACM*, pp. 739–747, 1982.

[14] C. Sminchisescu and B. Triggs, “Estimating articulated human motion with covariance scaled sampling,” *Int. Journal of Robotics Research*, vol. 6, pp. 371–379, 2003.