Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Visual Tracking Modalities for a Companion Robot

Paulo Menezes*, Frédéric Lerasle†, Jorge Dias*
*Institute of Systems and Robotics - Univ. of Coimbra, Polo II, 3030-290 Coimbra, Portugal, {paulo,jorge}@isr.uc.pt
†LAAS-CNRS, 7 av du colonel Roche, 31077 Toulouse cedex 4, France, lerasle@laas.fr

***Abstract*— This article presents the development of a human-robot interaction mechanism based on vision. The functionalities required for such mechanism range from user detection and recognition, to gesture tracking. Particle filters, which are extensively described in the literature, are well suited to this context as they enable a straight combination of several visual cues like colour, shape or motion. Additionally, different algorithms can be considered for a better handling of the particles depending of the context. This article presents the visual functionalities developed namely user recognition and following, and 3D gestures tracking. The challenge is to find which algorithms and visual cues fulfil the best, the requirements of the considered functionalities for our companion robot. The employed methods to attain these required functionalities and their results are presented.**

I. Introduction and framework

A major challenge, of the actuality, is undoubtelly the companion robot with the perspective of enabling a mobile autonomous machine to support modalities which are common in the interaction between humans. Gesture-based interaction is especially valuable in environments where the speech-based communication may be garbled or drowned out. Such interactions allow a robot companion to learn about the geometry and topology of the environments, the geometry, identity and location of objects, as well as their spatiotemporal relations. Once such companion robot has learnt, with the help of its tutor, all these informations, it can start interacting with its environment autonomously. In this context, we have designed and built a mobile robot named Rackham, a B21r robot made by iRobot, and which integrates some of the functionalities described in this paper. The visual interaction between humans and Rackham starts when it focuses its attention on specific persons (*i.e.* tutors), when they are detected on its vicinity.

Fig. 1. User interacting with Rackham.

Any person must be, normally identified before receiving the grant to interact with the robot. This requires the robot to keep checking on the detected persons until one is identified as a tutor. The maintenance of the interaction link requires that an identity verification step be executed repeatedly, to avoid that the robot commutes its attention from the current person to any other person present in its neighbourhood.

Regarding a key-scenario of H/R interaction, we consider that the tutor, after being identified as so, orders the robot to follow him. The robot complies, thanks to its basic mobility and visual analysis abilities. During this following task, the robot has to coordinate its displacements, even if only coarsely, with those of the tracked user, without being distracted by other people. Once the desired place is reached, the user may signal the mission end by a 'halt' gesture.

The approaches dedicated to these two first modalities, although providing only a coarse tracking granularity, are fast and robust. Another modality allows an active interaction with the robot by using not only communicative but also deictic gestures. The first type may be used to create a lexicon normally associated with commands, while the second one may offer an efficient modality to transmit information to the robot about the environment it evolves in. For creating this gesture interface, we need to perform the 3D tracking of the user's limbs.

Mobile robot applications impose several requirements on the developed modalities. First, the sensors setup is assumed to be embedded in the robot and so they are usually not static but moving, and their viewing field is quite narrow. All our visual modalities are based on a single colour camera mounted on our robot. Moreover, on-board computational power is limited and care must be taken to design efficient algorithms. As that the robot is expected to evolve in environments which are highly dynamic, cluttered, and frequently subjected to illumination changes, several hypotheses must be handled simultaneously. This is due to the multi-modality in the distributions of the measured parameters, as a consequence of the clutter or changes in the clothing appearance of the targeted subject. To cope with this, the integration of complementing visual cues is required.

Particle filtering is well-suited to this context, as it makes no restrictive assumptions about the probability distributions and enables the fusion of diverse measurements in a simple way. Although this fact has been acknowledged before [1], it has not been fully exploited in visual trackers. Combining a host of cues may increase the tracker versatility and reliability in our robotic context. This is achieved according to different schemes *e.g.* Condensation [2], I-Condensation [3] and Auxiliary [4]. Some of the variants are expected to fit to the requirements of the different modalities that compose the Rachkam interaction mechanism.

The paper is organized as follows. Section II presents the overall architecture of the interaction mechanism. Sections III and IV depict the tracking setups dedicated to the two first modalities *i.e.* the user identification and the robot guidance. Regarding the gesture-based interaction, section V details our approach for the 3D tracking of the upper human limbs and presents the results obtained for two test sequences. All the

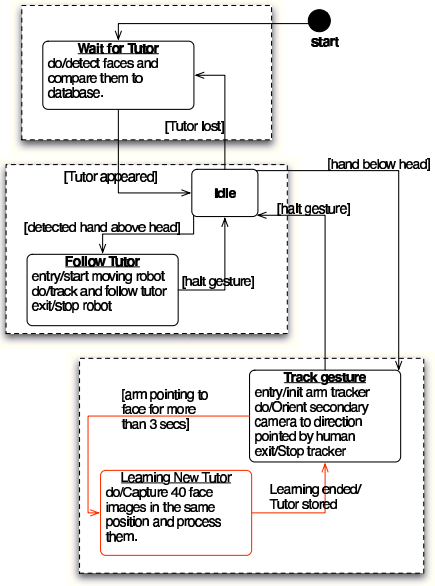


Fig. 2. States of our active H/R interaction scheme

implemented trackers were evaluated using a set of sequences acquired from the robot in a wide range of conditions like: cluttered environments, illumination variations, appearance variability of the targeted subject. Section VI summarizes our contribution and opens the discussion for future extensions.

II. Architecture

Figure 2 presents the functioning of the overall system, where three main parts can be clearly identified. The first one is dedicated to user face detection and identification. The system remains in this state until a known tutor appears. This event generates a transition to a "waiting" or idle state. This new state continues to verify the presence of the tutor and tests the input image for the presence of a hand in the open upright position. The relative position of the detected hand and the head defines the transition to the "tutor following" state or the "gesture tracking" state. In the former, the robot has to move following the user, and in the latter it tracks the user's gestures what can be used for communicate orders or point an object, a feature or another user to be leant. These functionalities are described in the following sections.

III. User Face Recognition and Tracking

Aiming to identify or confirm the identity of the person that is in the vicinity of the robot this module is composed of three parts, depicted hereafter and which are: face detection, face recognition and face tracking.

Face Detection: The method used for face detection was introduced by Viola *et al.* [5] and is based on a boosted cascade of classifiers built on Haar-like features. This detector relies on the relative contrast between some anatomical parts like the eyes and nose/cheek or nose bridge. The cascade of classifiers behaves as a degenerated decision tree where each stage contains a classifier which is trained to detect all frontal faces and reject only a small fraction of non-face patterns. At the end of the cascade, we can expect that "almost" all the non-face regions have been rejected, retaining for sure those containing faces. Figure 3 shows some examples where the rectangles outline the detected faces. The coordinates of these rectangles are are fed to next processing stage which performs the user recognition.

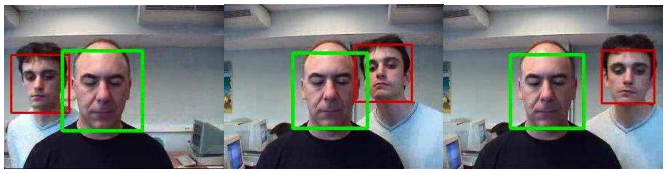


Fig. 3. Three frames from the face recognition output

Face Recognition: The face recognition step is based on the eigenfaces method introduced by Turk *et al.* [6]. Eigenvector-based methods are used to represent the learnt faces using low-dimensional vectors and then make it adequate both for storage and processing proposes. The Karhunen-Loeve Transform (KLT) and Principle Components Analysis (PCA) are the eigenvector-based techniques we use for dimensionality reduction and feature extraction in this automatic face recognition. Although this is a fast method, it imposes that every treated image be of the same size, and that all the objects to occupy most of that image. This is where the face detector comes to action, as it outputs the windows containing faces in the input image. Thus the subsequent step is to rescale each of these windows to the required size before passing them to the recognition step.

The combined face recognition system shows good results and acceptable processing times for eigenspaces created with 20 eigenimages. Figure 3 shows an example where the face detector marks the two faces but only one (marked as green) is recognised.

Face Tracking: During normal operation, more than one authorised tutor can be in the close to the robot, what could make system continuously switch from one to another. To avoid this, the eigenface descriptor of the selected user is used to verify that the interaction link is kept with the user that initiated it. The very sensitive nature of the face detector reflects itself in the production of false negatives. This can be filtered out by approximating the user's motion by a constant velocity model in a Kalman filter. In this case, when the face is not detected and consequently not recognised, the used motion model enables to predict its position in the image. One additional advantage of using the Kalman filter in this context is that, the face recognition can be accelerated. This is obtained by performing the face detection, which is the more computational demanding step, only on a region centred on and limited by the estimated position and covariance, respectively [7].

A recent tracking variant, which is well suited for this modality is proposed in [8]. This is based on I-Condensation scheme where importance sampling offers a mathematically principled way of directing search according to the face detector output. Regarding the particles weights definition, the measurement model fuses shape and color cues in the global likelihood (1).

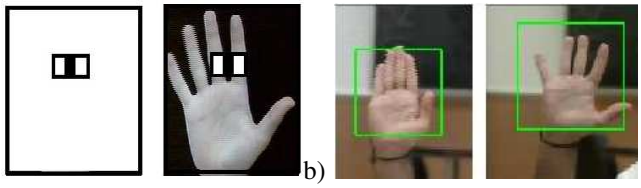


Fig. 4. a) One Haar feature detected on a hand. b) Output of the hand detector

Hand Detection using the Haar detector: The Haar-feature based algorithm was also successfully tested as a hand detector. For this, the classifier was trained with 2000 images containing upright hands, and 6000 images without hands and used as negative samples. This detector exhibits a detection rate slightly smaller than the previous, mainly due to the lack of discriminant contrasts in the hand. Figure 4a) shows the first feature detected on a hand, and figure 4b) shows some examples of hand detection. These results show that the obtained detector is able to cope with some deviance in hand orientation from the vertical. The detection of the open hand in this work is used to trigger the transition from the current state to a new one. So, accordingly to what is show in the schema of figure 2 once a hand is detected while in the "idle" state, the head detector is launched to find their relative positions to select the next modality to use.

IV. User tracking dedicated to tutor following

Regarding the guidance task, the robot has to coordinate its displacements with those of the guiding user. For this, the robot has to track the latter's motion, although not requiring a great precision. Once started, this tutor following modality runs until a open hand is detected, what can be interpreted as a "halt" sign.

To perform the required user tracking, we use the Condensation algorithm, as it is well adapted and permits a simple implementation. This estimates the state vector $\overrightarrow{x} = [x, y, \theta, s]^{'}$ which is composed of the position, orientation, and scale of the target in the image. The used measure is based on colour cues as they seem to be well-suited for this, even if we have to handle the appearance changes due to illumination variations, out-of-plane rotated faces, or robot motions. To overcome these appearance changes, we perform the update of the target's color model, allowing the on-line integration of limited variations of the observed characteristics with respect to the current reference model [9].

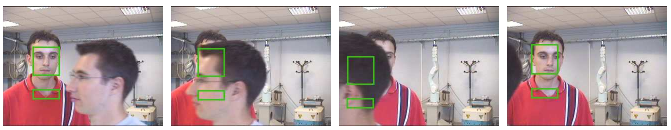


Fig. 5. Influence of the multi-part color model in the tracker

Dynamically updated models in trackers can lead to drifts with the consequent loss of the target. On the other side, with fixed model, the target's appearance can change due to any of the above mentioned reasons, and the target loss is inevitable as the model stops corresponding to the observations [1]. Some strategy is, by consequence, required to perform model update and ensure that model drift will not occur.

The used solution consists in using models composed of multiple colour-patches which are combined with shape cues in the computation of global likelihood (1) needed to the particle weighting step. Figure 5 shows some snapshots from a tracking sequence which includes temporary occlusions. If a single colour patch was used, the tracker would naturally adapt and lock to a wrong target that passes in the foreground. By using a multi-patch model, we ensure that, the tracker keeps locked onto the correct target even after the occlusion.

V. 3D Tracking for Gesture-Based Interaction

Gestures are commonly used to communicate or to simplify the communication between people. Consequently they appear as an excellent to transmit orders to a robot, or to refer to objects, locations, etc.

In our case, a particle filter-based tracker using a single camera as the information source, estimates the configuration of arms. The configuration vector, which represents a point in the 8-dimensional configuration space, cannot be compared directly to the images. Instead of that, the estimation is based on the observed appearance of the tracked subject in the input the images. Unfortunately, this measure-state link presents strong non-linearities due, not only to the projective projection process, but also to ambiguities produced by partial concealing that occur between body parts.

This tracking process, can be viewed as the iterative minimisation of a dynamic cost function, that evolves as the input view of the target changes over time. Its robustness depends, by consequence, on the shape of this cost function. If it presents multiple peaks, the tracker may be attracted to the wrong one with the consequent target loss, and if we succeed in making it unimodal or exhibiting a strong peak around the true point of the configuration space, the tracker will behave more robustly. One additional advantage of the particle filters is that even if the true shape of this cost function is not available, it can still be used as long as its value can be evaluated for any given point of the configuration space.

Robust cost function The cost function employed is a combination of several image measures related to the model and to some parameters that encode prior knowledge about the model or its physical properties. Used in the weighting step of the particle filter, this function is, by definition, proportional to the following probability density, $p(z|x)$, which represents the likelihood of the observed measure z given the configuration x. Considering that it is the combination of a set of M measures obtained from independent sources $(z_k^1, \ldots, z_k^M)$, it can be factorised as

$$p(z_k^1, \ldots, z_k^M|\mathbf{x}) \propto \prod_{m=1}^{M} p(z_k^m|\mathbf{x}). \quad (1)$$

The following subsections detail the various factors employed in the used cost function and which are related to image based measures and to physical properties of the model.

Shape cues: In our context, coarse 2D ou 3D models of the targeted limbs can be used. In a simple view-based shape representation, the limbs can therefore be rep-

resented by coarse silhouette contours (see figure 6) which result from the projection of a 3D model as described elsewhere [10]. This kind of model, although simplistic, permits to reduce the complexity of the involved computations. Indeed, this estimation process requires a preliminary 3D model projection with hidden parts removed. The associated likelihood is computed using the sum of the squared distances between model points and the nearest image edges [2]. The use of a Distance Transform, noted I_{DT}, obtained from the edges of the input image enables to avoid the search for edges in the neighbourhood of the projected contours. In addition to reduce the computational load, the use of the DT provides a smoother function of the model parameters.

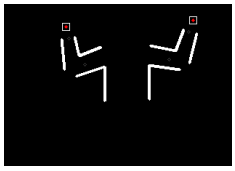
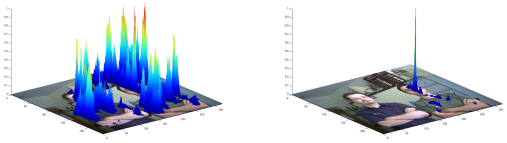


Fig. 6. Example of silhouette contours used as template

The edge image is here 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 will be a smoother function of the model pose parameters. Moreover, this reduces the involved computations because the DT image can be computed only once independently of the number of particles used in the filter. The edge-based marginal likelihood $p(z_k^S|\mathbf{x})$ is then given by

$$p(z_k^S|\mathbf{x}) \propto \exp\left(-\frac{D^2}{2\sigma_s^2}\right), \; D = \sum_{j=0}^{N_p} I_{DT}(j), \qquad (2)$$

where j indexes the N_p model points uniformly distributed along each visible model projected segments and $I_{DT}(j)$ the associated value in the DT image. Figure 7.(a) plots this function for an example where the target is a 2D elliptical template corresponding coarsely to the head of the right subject in the input image. As it can be seen on this plot, for cluttered background, using only shape cues for the model-to-image fitting is not sufficiently discriminant, as multiple peaks are present. Figure 8 shows the plot of the shape based likelihood function obtained by sweeping a subspace of the configuration space formed by 2 parameters of a human arms model. This plot's shape shows that this measure is not discriminant enough and so other measures are needed to remove the existing ambiguities.



(a) (b)

Fig. 7. Likelihoods regarding: (a) shape cue, (b) combined shape and motion.

Motion cues: In this context as the robot remains static during the gesture interaction, the used assumption is that the tutor arms are moving in front of a static background.

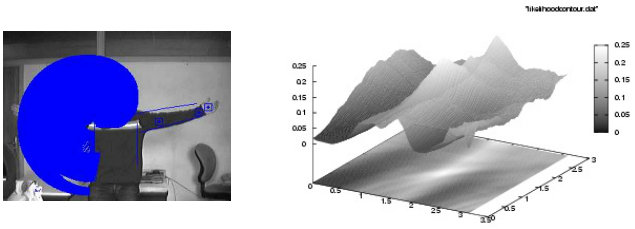


Fig. 8. Shape based likelihood obtained by sweeping the configuration subspace formed by 2 parameters of a human arms model

This allows to cope with cluttered scenes and reject false background attractors, by favouring the moving edges, as they are expected to correspond to the moving target. As the target can be temporarily stopped, the static edges are not completely rejected, but only made less attractive than the moving ones. 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 the local the optical flow vector $\vec{f}(z)$. From (2) and given K a constant, the new distance D is given by $D = \sum_{j=0}^{N_p} \min\left(I_{DT}(j), K.I'_{DT}(j)\right)$. Figure 7.(b) plots this more discriminant likelihood function for the example seen above. The results show that the tracking is less disturbed by the background clutter, especially while the target is moving.

Color cues: Reference colour models can be associated with the targeted ROIs. We denote the B-bin reference normalized histogram model in channel $c \in \{R, G, B\}$ by $h^c_{ref} = (h^c_{1,ref}, \ldots, h^c_{N_{bi},ref})$. The colour distribution $h^c_x = (h^c_{1,x}, \ldots, h^c_{N_{bi},x})$ of a region B_x corresponding to any state x is computed as $h^c_{j,x} = c_H \sum_{u \in B_x} \delta_j(b^c_u), j = 1, \ldots, N_{bi}$. $b^c_u \in \{1, \ldots, N_{bi}\}$ denotes the histogram bin index associated with the intensity at pixel u in channel c of the colour image, δ_a terms the Kronecker delta function at a, and c_H is a normalisation factor. The colour likelihood model must be defined so as to favour candidate colour histograms h^c_x close to the reference histogram h^c_{ref}.

From (2),the likelihood $p(z^C_k|\mathbf{x})$ is based on the Bhattacharyya coefficient [1] between the two histograms h^c_x and h^c_{ref}. $D(h_x, h_{ref}) = (1 - \sum_{j=1}^{B} \sqrt{h_{j,x}.h_{j,ref}})^{1/2}$.

The smaller D is, the more similar the distributions are. Finally, the likelihood model used $p(z^C_k|\mathbf{x})$ is given by $p(z^C|x) \propto \exp(- \sum_{c \in \{R,G,B\}} D^2(h^c_x, h^c_{ref})/2\sigma^2_C)$.

Figure 9 plots an approximation of the likelihood function, based on the comparison of the colour distributions using the Bhattacharyya coefficient. The reference colour ROI corresponds to the right hand of the person as marked on the image. The plot is obtained by computing the Bhattacharyya distance between the colour distribution of the reference ROI and a patch swept allover the image.

From this measure, we can also define a likelihood $p(z^T_k|\mathbf{x})$ relative to textured patches based on the intensity component.

The results of combining this colour based likelihood with the previously presented one based on template matching are shown in figure 10.

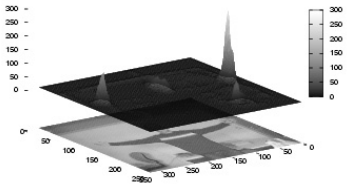



Fig. 9. Likelihood surface obtained by comparing the colour distribution of a patch taken from the right hand which the whole image

Once again, this was obtained by sweeping the configuration subspace of a human arms model corresponding to the parameters of two joints: right arm's elbow and shoulder fronto-planar rotation. This combined result shows a significant improvement in terms of the likelihood function's shape. In this case a quite pronounced peak appears around the true configuration.

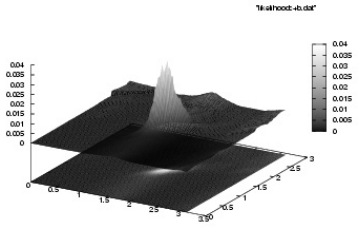


Fig. 10. Likelihood function using shape and colour measures, for the case of figure 8

Non-observable parts stabilisation: Despite the visual cues depicted above, ambiguities arise when certain model parameters cannot be inferred from the current image observations, especially for a monocular system. They include, but are not limited to, kinematic ambiguities. For instance, when one arm is straight and the edge-base likelihood (2) is used, rotation of the upper arm around its axial axis is unobservable, because the model projected contours remain static under this DOF. Leaving these parameters unconstrained is questionable. For this reason, and like in [11], we control these parameters with a stabiliser cost function that reaches its minimum on a predefined resting configuration $\mathbf{x}_{def}$. This enables the saving of computing efforts that would explore the unobservable regions of the configuration space. In the absence of strong observations, the parameters are constrained to lie near their default values whereas strong observations unstick the parameters values from these default configurations. The likelihood for state $\mathbf{x}$ is defined as: $p_{st}(\mathbf{x}) \propto \exp(-\lambda_{st}||\mathbf{x}_{def} - \mathbf{x}||^2)$.

This prior only depends on the structure parameters and the factor λ_{st} is chosen so that the stabilising effect will be negligible for the whole configuration space with the exception of the regions where the other cost terms are constant.

Collision detection: Physical consistency imposes that the different body parts do not inter-penetrate. As the estimation is based on a search on the configuration space it would be desirable to a priori remove those regions that correspond to collisions between parts. Unfortunately it is in general not possible to define these forbidden regions in closed form so they could be rejected immediately during the sample phase. The result is that in the particle filter framework, it is possible that configurations proposed by some particles correspond to such impossible configurations, thus exploring regions in the configuration space that are of no interest. To avoid these situations, we use a binary cost function, that is not related to observations but only based on a collision detection mechanism. The likelihood function for a state $\mathbf{x}$ is $p_{coll}(\mathbf{x}) \propto \exp(-\lambda_{co} f_{co})$ with: $f_{co}(\mathbf{x}) = \left\{ \begin{array}{ll} 0 = \text{No collision} & 1 = \text{In collision} \end{array} \right\}$

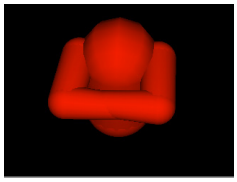


Fig. 11. Self-collision configuration proposed by a particle

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, it avoids the drifting of the filter to zones of no interest, and it avoids wasting time in performing the measuring step for unacceptable hypothesis.

Implementation: In its actual form, the system tracks the parameters of a model containing eight degrees of freedom, *i.e.* four per arm as shown in figure 12. 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/scale of the face given by the face detector of the previous step. In addition to the projected contours of the model, a set of colour 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.

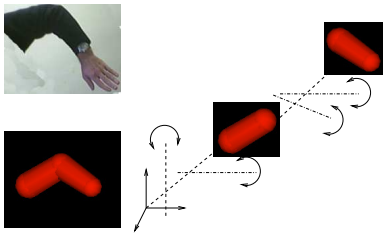


Fig. 12. Arm structure exhibiting its DOF.

Regarding the particle filtering framework, we opt for the Auxiliary Particle Filter scheme [4], 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 associated measurement strategy is as follows: (1) particles are firstly located in good places of the configuration space according to rough correspondences between model patches and image features, and (2), on a second stage, particles' weights are fine-tuned by adding edges cues, motion information, etc.

Due to the robotics requirements, our tracker must adapt automatically to the variabilities of both the clothing appearance and environmental conditions. Therefore, some heuristics allows to weight the strength of each visual cue in the global likelihood (1). 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 reference histograms h^c_{ref} and h^I_{ref} are uniform so that for the colour $h^{c,I}_{j,ref} = \frac{1}{N_{bi}}$, $j = 1, \ldots, N_{bi}$. Typically, uniform coloured patches produce low likelihood values, whereas higher likelihood values characterise confi-

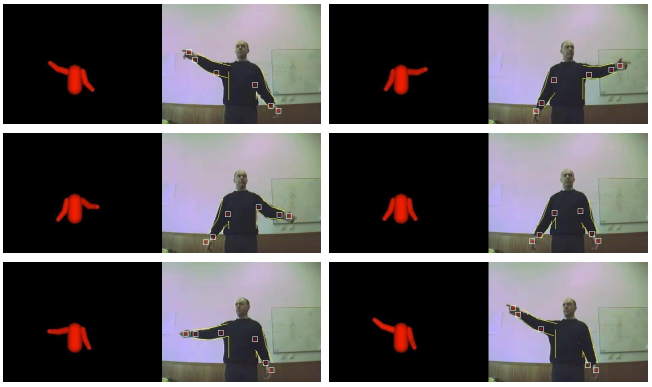


Fig. 13. From top-left to bottom right: snapshots of tracking sequence (pointing gestures)

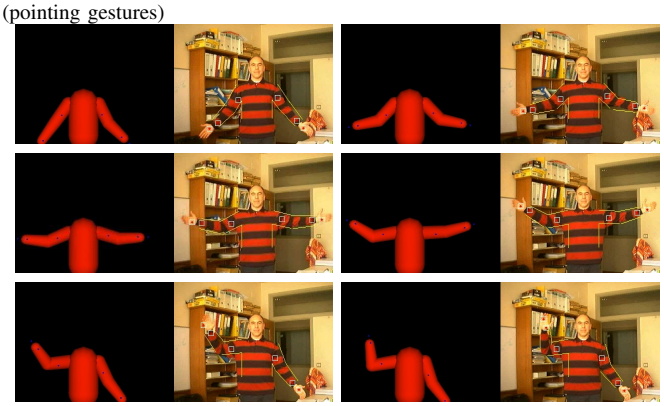


Fig. 14. From top-left to bottom right: snapshots of tracking sequence (heavy clutter)

dent patches because their associated colour distributions are discriminant and ensure non ambiguous matchings. By this way, parameter λ_p weights the strength of the $p-th$ marker in the likelihood function (1). Parameter λ_s weights the edges density contribution and is fixed from the first DT image of the sequence.

The above described approach has been implemented and evaluated over monocular images sequences acquired in various situations. Figure 13 shows snapshots of the results obtained from one of the evaluation sequences. The right sub-figures show the model projections superimposed to the original images for the mean state $E[\mathbf{x}_k^i]$ at frame k, while the left ones show its corresponding estimated configuration. These examples combine measures that use the projected contours, three patches per arm, and the geometric constraints.

For this first scenario (figure 13), that shows the tracking of pointing gestures, the target contours are prominent and are weakly disturbed by the background clutter. The high confident contours cue ensure the tracking success. This permitted the tracking of the arms even when they got out of the fronto-parallel plane thanks to all the patches (figure 13).

For cluttered background the gestures tracking is also performed successively, as shown in figure 14. This scenario clearly takes some benefits from the discriminant patches and from the use of optical flow which weights the importance relative to the foreground and background contours. If considering only contour cues in the likelihood, the tracker would attach itself to cluttered zones and consequently lose the target.

Other experiments, available at the authors' webpage (`http://www.isr.uc.pt/~paulo/HRI`), demonstrate the tracker's ability to follow a wide range of two arms movements despite very strong variability in shape and appearance due to both arm muscles and clothing deformations.

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.* $N_s = 400$ particles. In our unoptimised implementation, a PentiumIV-3GHz requires about $1s$ per frame to process the two arm tracking, most of the time being spent in observation function. To compare, classic systems take a few seconds per frame to process a single arm tracking.

VI. Conclusion

This article presents the development of a set of visual functions that aim to fulfil a basic step of interaction functionalities. Face detection and recognition based on Haar functions and eigenfaces enable the recognition of the tutor users. A modified Haar-based classifier was created to detect open hands in images. User tracking to make the robot follow the user is implemented using a particle filter that uses colour distribution over rectangular patches as target features. The colour distributions that correspond to each patch are updated on-line to the changes produced by the targets motion or illumination changes. Finally a method capable of tracking the configuration of the human arms from a single camera video flow is presented. Future works include the optimisation of the 3D tracker so it can be used in realtime video flows, enabling it to be used interactively to communicate with the robot.

References

[1] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *IEEE*, vol. 92, no. 3, 2004.

[2] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *ECCV'96*, Cambridge, UK, April 1996, pp. 343–356.

[3] ——, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," in *ECCV'98*, 1998, pp. 893–908.

[4] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, 1999.

[5] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *CVPR'01*, 2001.

[6] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *CVPR'91*, 1991, pp. 586–591.

[7] P. Menezes, J. C. Barreto, and J. Dias, "Face tracking based on haar-like features and eigenfaces," in *IAV2004*, Lisbon, Portugal, July 5-7 2004.

[8] L. Brthes, F. Lerasle, and P. Dans, "Data fusion for visual tracking dedicated to human-robot interaction," in *ICRA'05*, 2005.

[9] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptative color-based particle filter," *Journal of Image and Vision Computing*, vol. 21, no. 90, pp. 90–110, 2003.

[10] P. Menezes, F. Lerasle, J. Dias, and R. Chatila, "Appearance-based tracking of 3D articulated structures," in *ISR2005*, Tokyo, Japan, November 2005.

[11] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *IEEE Int. Journal on Robotic Research* , vol. 6, no. 22, pp. 371–393, 2003.