Visual Tracking Based Modalities Dedicated to a Robot Companion

Paulo Menezes[1], Frédéric Lerasle[2], and Jorge Dias[1]

[1] ISR/DEEC - University of Coimbra, Coimbra, Portugal
[2] LAAS-CNRS, Toulouse, France
{paulo,jorge}@isr.uc.pt, 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. The employed methods to attain these required functionalities are described with results presented.

1 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) who are detected and identified in its vicinity. The interacting person must be identified before receiving the grant to interact with the robot.

Figure 1. User interacting with Rackham

This requires to keep track on the detected persons while one is not identified as a possible tutor. Maintaining the interaction link requires that an identity verification step be executed repeatedly, otherwise the robot could jump its attention from the current person to any other person present in its neighbourhood.

As an example of interaction with the robot, the identified tutor can order the robot to follow him. The robot complies, thanks to its basic mobility and visual analysis abilities. During the guidance task, the robot has to coordinate its displacements, even if coarsely, with the tracked user, without being distracted by other people. Having reached the desired place, the user signals the mission end by a 'halt' gesture. These two modalities are developed using approaches that, although providing only a coarse tracking granularity, are fast and robust.

The user can then interact actively with the robot using not only to communicative but also deictic gestures as they constitute an efficient modality to transmit information to the robot about the environment around it. For this, as will be described later in this article, we perform the 3D tracking of the user's limbs using a single camera.

Our strategy uses a single view of the person to track, what is made possible by increasing the reliability and specificity of the observation model. This aims to deal with some of the requirements of a mobile robot. First onboard sensors are moving and with limited field of view. 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, a robust integration of multiple visual cues is required.

Particle filtering seems to be 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 [8], it has not been fully exploited in visual trackers, for which a general review can be found in [2]. Combining a host of cues may increase the tracker versatility and reliability in our robotic context. Some of the variants of the particle filter are expected to fulfill to the requirements of each the different modalities that compose the Rackham interaction mechanism.

The paper is organized as follows. Section 2 presents the overall architecture of the interaction mechanism. Sections 3 and 4 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 5 details our approach for the 3D tracking of the upper human limbs and presents the results from one of the test sequences. Last, section 6 summarizes our contribution and opens the discussion for future extensions.

2 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.

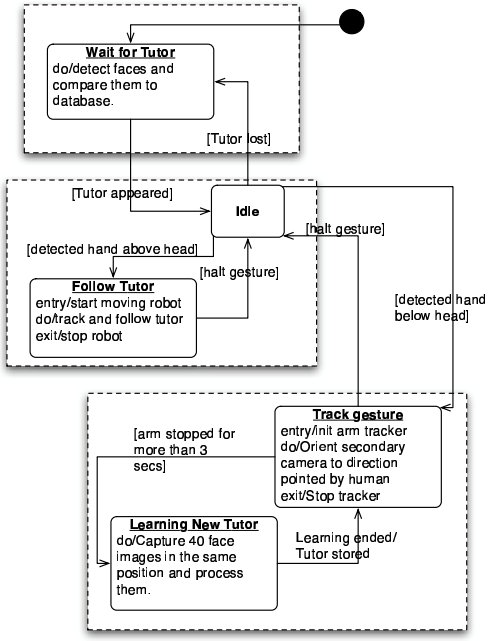

Figure 2. States of our active H/R interaction scheme

3 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.* [12] 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.

Face Recognition: The face recognition step is based on the eigenfaces method introduced by Turk *et al.* [11]. 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 Principal 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. The combined face recognition system shows good results and acceptable processing times for eigenspaces created with 20 eigenimages. Figure 3 shows three frames of a given sequence.The face detector marks the three faces but only one (marked as green) is recognised as corresponding to the previously learnt eigenspace.

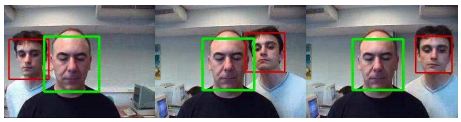

Figure 3. Three frames from the face recognition output

Face Tracking: : As another authorised tutor can be in the neighbourhood and the system could switch from one to the other. 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 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. The face recognition can then be accelerated by performing the face detection only on a region centred on and limited by, the estimated position and covariance, respectively [5].

Another possibility is the use of the output of the face recognition to initialise a particle filter prior, or to use it in the importance function as in [1] for a I-Condensation implementation [4].

Using the Haar Detector to Detect Hands: 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 slightly smaller detection rate, than the previous, due to the lack of discriminant contrasts in the hand. Figure 4 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 change from the current state to a new one. So 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.

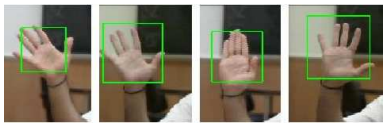

Figure 4. Output of the hand detector

4 User tracking dedicated to tutor following

Regarding the guidance task, the robot has to coordinate its displacements with the guiding user. Although not requiring an important precision, the robot has to track the latter's motion. 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 conditions, out-of-plane rotated faces, and 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 [7].

One must be aware that the use of dynamically updated models in trackers can lead to drifts with the consequent loss of the target. On the other side, if a fixed model is used and the target's appearance can change due to its own motion, to the variations of the illumination conditions, or to any other reason, the target loss is inevitable as the model stops corresponding to the observations [8]. A strategy is by consequence required to perform model update and ensure that model drift will not occur.

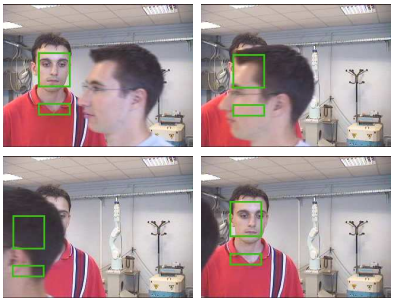

Figure 5. Influence of the multi-part color model in the tracker

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 sequence including temporary occlusions. With a single colour patch, the tracker would naturally adapt and lock to a wrong target that passes in the foreground. When using a multi-patch model, the tracker keeps locked onto the correct target even after the occlusion.

5 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 form 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 inherent to the projective projection process but also due to ambiguities produced by partial concealing that occur between body parts. The particle filter is quite adapted to these situations as it can handle not only nonlinear models but also non-Gaussian distributions.

Contrary to the Kalman filter, where the state distribution is represented by a mean and covariance, the particle filter represents this by a set of weighted samples. For the current case, each sample represents an hypothetic joint configuration of the two arms. Its weight is then computed by obtaining the projection of the model corresponding to this particle, and then compare the result to the input image. Both the construction of the model for the arms model based on quadrics and the generation of its projection is described in [6].

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.

5.1 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

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

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 represented by coarse silhouette contours. 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 [3]. 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.

The edge-based marginal likelihood $p(z_k^S|\mathbf{x})$ is then given by

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

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.

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. 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 $\mathbf{f}(z)$. From (2) and given K a constant, the new distance D is given by

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

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 $h_{ref}^c = (h_{1,ref}^c, \ldots, h_{N_{bi},ref}^c)$. The colour distribution $h_x^c = (h_{1,x}^c, \ldots, h_{N_{bi},x}^c)$ of a region B_x corresponding to any state x is computed as $h_{j,x}^c = c_H \sum_{u \in B_x} \delta_j(b_u^c), j = 1, \ldots, N_{bi}$. $b_u^c \in \{1, \ldots, N_{bi}\}$ 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_x^c close to the reference histogram h_{ref}^c. From (2), the likelihood $p(z_k^C|\mathbf{x})$ is based on the Bhattacharyya coefficient [8] between the two histograms h_x^c and h_{ref}^c. This likelihood can be extended to consider to consider several patches of distinct colours, *e.g.* the limbs and clothes of a person. It suffices to split the ROI into subregions, each with its own reference colour model [8].

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

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 [10], 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 function for a state $\mathbf{x}$ is defined as:

$$p_{st}(\mathbf{x}) \propto \exp(-\lambda_{st}||\mathbf{x}_{def} - \mathbf{x}||^2). \tag{3}$$

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 other cost terms are constant.

Collision detection: Physical consistency imposes that the different body parts do not interpenetrate. 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:

$$f_{co}(\mathbf{x}) = \begin{cases} 0 \text{ No collision} \\ 1 \text{ In collision} \end{cases}$$

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 derivation 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.

Implementation: In its actual form, the system tracks the parameters of a model containing eight degrees of freedom, *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/scale of the

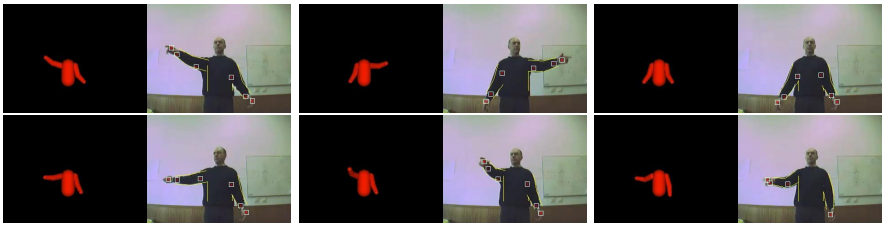

Figure 6. From top-left to bottom right: snapshots of tracking sequence (pointing gestures)

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.

Regarding the particle filtering framework, we opt for the Auxiliary Particle Filter scheme [9], 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.

The above described approach has been implemented and evaluated over monocular images sequences acquired in various situations. Figure 6 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. The following examples combine measures that use the contours, three patches per arm, and the previously described geometric constraints.

Other experiments, available at the authors' webpage, 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.

6 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. L. Brethes, F. Lerasle, and P. Danes. Particle filtering strategies for visual tracking dedicated to H/R interaction. In *Workshop on Human/Robot Interaction*, Submission, Palerma, 2006.
2. D.M. Gavrila. The visual analysis of human movement : A survey. *Computer Vision and Image Understanding (CVIU'99)*, 73(1):82–98, 1999.
3. M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Europ. Conf. on Computer Vision (ECCV'96)*, pages 343–356, Cambridge, UK, April 1996.
4. M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Europ. Conf. on Computer Vision (ECCV'98)*, pages 893–908, 1998.
5. Paulo Menezes, José Carlos Barreto, and Jorge Dias. Face tracking based on haar-like features and eigenfaces. In *5th IFAC Syposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 5-7 2004.
6. Paulo Menezes, Frédéric Lerasle, Jorge Dias, and Raja Chatila. Appearance-based tracking of 3d articulated structures. In *36th International Symposium on Robotics*, Tokyo, Japan, November 2005.
7. K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptative color-based particle filter. *Journal of Image and Vision Computing (IVC'03)*, 21(90):90–110, 2003.
8. P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *IEEE*, 92(3), 2004.
9. M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446), 1999.
10. C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *IEEE Int. Journal on Robotic Research (IJRR'03)*, 6(22):371–393, 2003.
11. M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'91)*, pages 586–591, 1991.
12. P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, 2001.