Visual tracking of silhouettes for human-robot interaction

P. Menezes†,* L. Brèthes‡, F. Lerasle‡, P. Danès‡, J. Dias†

†ISR/DEEC - Univ. Coimbra
Pinhal de Marrocos
3030-290 Coimbra - Portugal

‡LAAS-CNRS
7, av. du Colonel Roche
31077 Toulouse Cedex 4 - France

{pmenezes,lbrethes,lerasle,danes}@laas.fr, jorge@isr.uc.pt

Abstract

The interaction between man and machines has become an important topic for the robotics community as it can generalise the use of robots. One of the requirements for this interaction is that a robot be able to detect and analyse the motion of a person in its vicinity. This paper describes a Monte-Carlo based method for human head/hand detection and tracking in video streams. The tracked part is modelled by a spline. The pose estimation consists in fitting the model to the current image gradient taking into account with motion measurements. Results of detection and tracking using these combined criteria are illustrated. The limits of the method are also discussed. Finally, future extensions are proposed, based on colour segmentation, to improve the robustness of the method.

1 Introduction

Man-machine interaction has become an important topic in the robotics community. In this context, advanced robots must integrate capabilities to detect humans presence in their vicinity and interpret their motion. This permits to anticipate and take countermeasures against any possible collision or passage blockage.

For an active interaction, the robot must also be able to follow a person's gestures, as they can be part of an object exchange or of a communication process. This requires the determination of the person's pose as well as the location of the hand(s) and the estimation of their trajectories.

Many researchers have successfully tried to detect and track people or human parts in video streams from one or more cameras. Some of the existing approaches use prior models of the human body (or body parts) and/or make assumptions on the motion characteristics [5] to be detected and analysed. These models are either 2D [1] (image plane models) or 3D [9] (wire-frame or solid shape structures) and can be deformable [4] or rigid [9].

Our approach is based on coarse 2D rigid models of the human head or hand. These models although simplistic permit to reduce the complexity of the involved computations and still obtain good results as will be shown later.

Section 2 describes the method and focuses on motion considerations to detect and track a human head or hand. Results are presented in section 3. The performance and limitations of the approach are also discussed therein. Regarding the limitations, some considerations are outlined on how to integrate colour information to improve the robustness.

2 Visual tracking method

2.1 Overview

Performing automatically the visual tracking of a given object is a very valuable goal but, unfortunately, it still does not have a general answer. The fact is that an object can generate very different images depending on its pose or illumination. Silhouette-based approaches simplify the problem by reducing the variability of the object representations.

In this case the silhouette contour is modelled by a spline whose state is estimated using a particle filter. Examples of these models are shown in figure 1. The choice of a particle filter as the tracking engine comes from its capability to work in the presence of nonlinearities and non-Gaussian noise models. The details of this filter and its associated formalism can be found in [6, 3, 1].

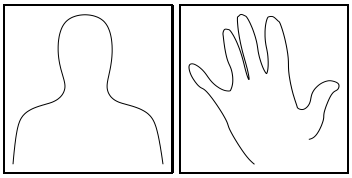


Figure 1: *Contour templates for head or hand*

Blake *et al.* [1] propose to consider the local deformation of the model by augmenting the state vector with the coordinates of the spline control points. However this substantially increases the state space dimension and thus deteriorates the performance of the particle filter. So, a four-

* Partially supported by grant from FCT-MCT

dimensional state vector is considered, composed of the position coordinates, the scale and the orientation of the target silhouette in the image. In the following, this state vector is noted $\vec{x} = [x, y, s, \theta]^T$.

2.2 Dynamic model

The target dynamics are depicted by the auto-regressive model

$$\vec{X}_k = \mathbf{A}\vec{X}_{k-1} + \vec{W}_k,\ with\ \vec{X}_k = \begin{bmatrix} \vec{x}_k \\ \vec{x}_{k-1} \end{bmatrix}$$

where k is relative to the kth image of the sequence and W_k terms the process noise. This model is used during the prediction step. It is worth noting that due to the versatility of the particle filter, the above dynamics could have been chosen nonlinear and the process noise could be non-Gaussian.

2.3 Measurement model

The measurement model links the system state with the measured output. In the particle filter update step, each particle must be weighted by the likelihood that relates the current measured output and state that corresponds to the particle.

In the present case, the likelihood of each sample depends on the sum of the squared distances between model points and corresponding points on the image. The model points are chosen to be uniformly distributed along the spline. The closest edge point in the normal direction at each sampled point of the spline is selected and the euclidian distance between these two points constitutes our local error measurement (figure 2). The matching criterion between the proposed model state and the image edges is defined in terms of conditional probabilies as

$$p(z_k|x_k^i) \propto \exp\left(-K\sum_{j=0}^{N}\phi(j)\right) \quad (1)$$

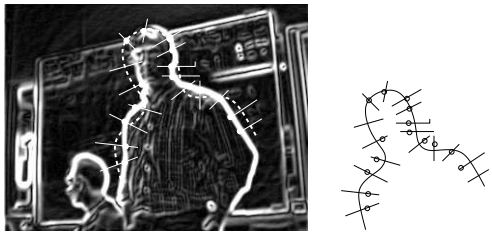
by setting $d(j) = |x_k^i(j) - z_k^i(j)|$, $\phi(j)$ is given by

$$\phi(j) = \begin{cases} d(j)^2 & \text{if } d(j) < \delta \\ \rho & \text{otherwise} \end{cases} \quad (2)$$

with $x_k^i(j)$ the j-th measurement point of the spline corresponding to the i-th particle at time k, $z_k^i(j)$ the closest edge point on the spline normal, K, ρ and δ predefined constants.

2.4 Improvements to the measurement model

The simple measurement model is shown to work quite well if the outer contour of the silhouette can be properly extracted and the background does not present much clutter that may confuse the tracker. Unfortunately this is not always true and using only edges for template fitting is not sufficient to make it robust enough. In some situations, due to unfavourable illumination, the target contour may not be so prominent as expected. Then there may be portions of the image in which the template matches better with irrelevant contours than with the ones of the true target. Consequently, the tracker may loose the target and "attach itself" to that cluttered zone. This phenomenon happens because the features used by the measurement model are not sufficiently discriminant. Having stated this, and assuming that the target is moving most of the time, the inclusion of motion based information in the likelihood expression (1) has been considered.



a) b)

Figure 2: *a) Measuring the correspondence between a particle and an edge image; b) A representation of the contour template, the spline normals and points that match image edges.*

Blake *et al.* described in [10] a tracker which combines Kalman filtering and background modeling i.e. a statistical form of background subtraction. In our case, background subtraction facilitates the tracker initialisation which remains often problematic, especially for cluttered background. A variant is to consider optical flow.

Optical flow, though marginally used by Blake *et al.*, allows to remove efficiently static clutter from the image data. The two next sections describe these two directions, namely the use of background subtraction and optical flow, especially for initialisation phase focused on section 2.5.

2.4.1 Background subtraction. The segmentation of a moving target out of a slowly-varying background can be set up by carrying pixel-wise computations on an intensity image. These lead to the labelling of each point as a member of either the background or the target. The interests of such a method are its ease of implementation, its low computational cost enabling it to run at video rate, and the fact that it doesn't require the use of a colour camera. Yet, errors are to be expected due to the labelling of each pixel regardless of its vicinity and of any *a priori* hypothesis on the target's motion type. Nevertheless, some of these false detections in the segmented binary image can be eliminated by subsequent morphological filtering.

As the considered video sequences are long, wide intensity variations often happen both on the background and on the target. The detection scheme proposed by Donohoe *et al.* in [2] is designed so as to be well-behaved even in such a context. It consists in labelling each image pixel from its intensity by applying a simple binary hypothesis test.

The distribution of the intensity of each image pixel has then to be characterised at every instant under each of the two hypotheses "background" and "target". As the target can hold enlightened and shaded zones depending on the scene illumination, its points intensities can extend from dark to bright. Said it mathematically, the conditional probability density function of the intensity of any pixel under the hypothesis that it belongs to the target is uniform whatever the considered image in the sequence. Besides, the intensities distributions of the points lying on the background entail the computation at each instant of a so-called dynamic reference frame, viz. of a secondary image whose pixels are all obtained separately by a discrete-time unitary static gain first-order filtering of the corresponding pixels throughout the sequence. Choosing a high enough time-constant leads to filter solely the moving target, so that the dynamic reference frame is a good estimate of the background. The difference between this image and the actual one can thus be assimilated as zero-mean additive imaging noise. To get the conditional probability density for an image pixel assuming this pixel belongs to the background, it is then sufficient to empirically compute the – zero-mean – spatial dispersion of the imaging noise over the whole sequence, and shift it in order to balance it on both sides of the intensity of the corresponding pixel in the dynamic reference frame.

Once they are evaluated for the actual intensity of the image pixel under categorisation, the above conditional probability density functions are used as likelihoods for hypotheses testing. As is usually done, they are compared by means of a threshold, which can be selected so that the decision process satisfies a predefined performance index [12].

Donohoe *et al.*' technique was originally designed in order to extract small-sized targets moving fast enough. In the context of human-robot interaction by gestures, the hand motion may be quite slow. Moreover, segmenting both the hand and the forearm, which may be worth to help the interpretation, can fail because of the forearm's quasi-immobility. The above technique has thus been implemented with a slight modification enabling its use for gestures segmentation.

Assume that i is a dummy variable representing an intensity, n is the total number of intensity levels, λ is the decision threshold, and, at time k, i_n terms the intensity of a pixel of the actual image, m_k terms the intensity of the corresponding pixel on the dynamic reference frame and σ_k is the standard deviation of the – presumably Gaussian – imaging noise. The computations concerning each image pixel at time k are as follows, the last one being somewhat different from the original method of Donohoe *et al.*:

1. state $p(i|\text{target}) = \frac{1}{n}$ and $p(i|\text{background}) \sim \mathcal{N}(m_{k-1}, \sigma^2_{k-1})$;
2. if $\frac{p(i_k|\text{background})}{p(i_k|\text{target})} > \lambda$ [resp. $< \lambda$], then decide the current pixel is a member of the background [resp. of the target];
3. if the current pixel is a member of the background, then state $m_k = \alpha i_k + (1-\alpha) m_{k-1}$ with $\alpha \in [0;1]$, else state $m_k = m_{k-1}$; repeat a similar filtering process for the computation of σ^2_k.

Various values of λ have been tried out. Let P_F [resp. P_M] be the probability to decide that the pixel under concern belongs to the background while it lies on the target [resp. belongs to the target while it lies on the background]. First, λ has been computed so as to minimise P_M with P_F fixed to 5%. The same was then done after reversing the roles of P_M and P_F. Finally, λ was set to 1, which turns to minimise the probability $P_M + P_F$ of making a wrong decision. As the image noise variance σ^2_k is always greater than 5 in our experimental context, the two first strategies give results very close to setting $\lambda = 0.1$. Moreover, if $\sigma^2_k < 20$, the decision differs from the third strategy only by up to 4 intensity levels. So, $\lambda = 1$ has been selected. Figure 3 presents an example of background estimation. Here the top row shows a sequence of 3 input images and the bottom row the corresponding evolution of the background estimate.

Figure 3: *Example of background estimation. Top: input video sequence; bottom: resulting background image update*

2.4.2 Optical flow. Being optical flow the apparent motion induced in the image sequence by the relative motion between the camera and the scene, it would permit the separation between the moving objects and the fixed background. Existing techniques to estimate optical flow vectors for every pixel in the image have been used. Nevertheless, this kind of information can only be used if considering that the camera is fixed or is undergoing a pure rotation with known angular velocity. In the latter case, the effects of the rotation can be estimated and then compensated on the computed vectors.

The optical flow field can then be used to create a mask which selects only the edges that correspond to the moving zone for the calculation of the likelihood function (1). Then, the points $z^i_k(j)$ in equation (2) receive the additional constraint that the corresponding optical flow vectors must have nonzero norm.

This mask selection permits the distinction between static background edges and the target moving edges. Nevertheless, this method makes the tracker fail as soon as the

person stops moving. So, instead of removing the edges that do not move, the moving ones are just more favoured, so that the tracker will prefer them if they exist but still finds the motionless ones. Given $\vec{f}(z_k^i(j))$ the optical flow vector for pixel $z_k^i(j)$, the expression (2) is then replaced by:

$$\phi(j) = \begin{cases} d(j)^2 + \rho\gamma(z_k^i(j)) & \text{if } d(j) < \delta \\ 2\rho & \text{otherwise.} \end{cases} \quad (3)$$

with

$$\gamma(z_k^i(j)) = \begin{cases} 0 & \text{if } |\vec{f}\left(z_k^i(j)\right)| \neq 0 \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

2.5 Initialisation

Filter initialisation is a crucial aspect of the tracking process. An accurate initial estimate is required unless the system can be fairly approximated by a linear model subject to Gaussian noise. For instance a bad initial estimate can make the filter converge to a local minimum other than the one corresponding to the target. Then, the particle filter should be initialised with a set of samples which can be generated from a multivariate Gaussian distribution centred on some fairly good initial state estimate X_0 and covariance Σ.

The proposed initialisation method consists once again in the use of motion information. So, the initial estimate is extracted *via* the detection of a moving zone in the image plane.

2.5.1 Initialisation using background subtraction. Background subtraction techniques can be used once again for the initialisation purpose. These allow to coarsely isolate the image area corresponding to the target. So, the research area in the image can be reduced and an initial estimate X_0 can be deduced from the inertial moment characteristics of the isolated object pixels. Figure 4 shows an example of hand segmentation using the method discussed in [2].

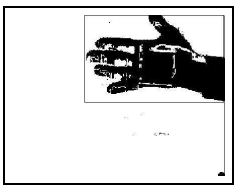


Figure 4: *Region of interest on an image issued from background subtraction*

2.5.2 Initialisation using optical flow. While the previous approach is restricted to fixed camera contexts, the optical flow based approaches allow the separation of zones with different directions of movement.

The target region, once isolated, is then used to assign an initial estimate to the tracker.

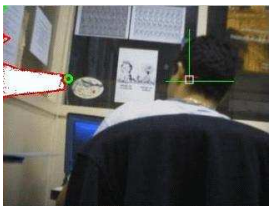


Figure 5: *Estimating the initial target location using motion information*

Considering the case of tracking the head and torso of a person and after having identified the moving area, the initial state estimate is obtained using the following assumptions:

- Normally people walk upright so the initial **orientation** θ is fixed and equal to $\theta_{upright}$.
- Detecting the head position is done by scanning the image from top to bottom to select the first pixel that has a non zero motion vector after noise filtering. The model can be described with respect to this point so the **position** part (x, y) of the vector state is known.
- The **scale** s can be inferred from the width of the lines of the moving zone. This is done by searching for the first local maximum value in the top-to-down width sequence.

As these estimated values are used to generate a multivariate Gaussian distribution, the corresponding positions in the diagonal of the covariance matrix must reflect the uncertainty associated to the initial estimate.

Figure 5 shows an example of obtaining an initial value before launching the particle filter for the case of tracking a head. As there is no automatic mechanism for inferring the variance values these are preset by hand.

It should be noted that for other target types, e.g. hands, a similar approximation can be applied, though it would require some constraints on the initial hand pose.

3 Results and future works

3.1 Tracking results

The presented method has been implemented on a PIII-1GHz laptop running Linux. Although no special care was taken in terms of code optimisation, it was possible to either track a hand or a head with reasonable performance. The original method performs quite well in the presence of some background clutter as can be seen on figure 6. Yet, the background can generate too many edges which may "confuse" the tracker as happens in figure 7. This happens because, contrarily to Blake et al., the template deformation, that could render it closer to the hand contour, is not allowed in this implementation. Nevertheless, by using a lower dimension state space, a better behaviour of the particle filter is expected.

Considering this, cluttered edges due for example to complex background may exhibit a better matching criterion with the template than the edge features which really belong to the real target.

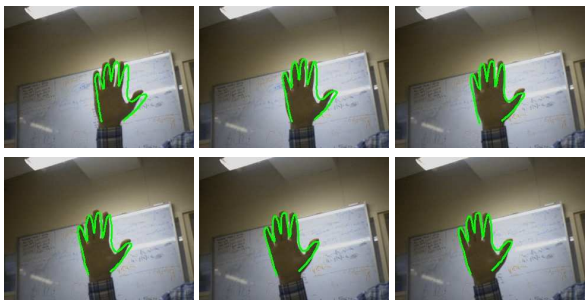


Figure 6: *Tracking example*

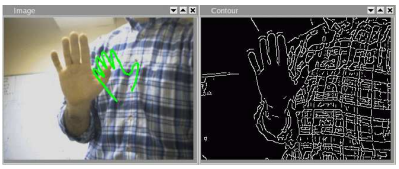


Figure 7: *Example case where the criterion based only on distance fails*

Adding the motion constraint to the tracker matching criterion has permited to improve its performance. Actually, this made it converge in situations where it would normally fail (figure 8). Figure 9 also shows a tracking example where the used spline roughly models the shape of the head-neck-shoulders set.

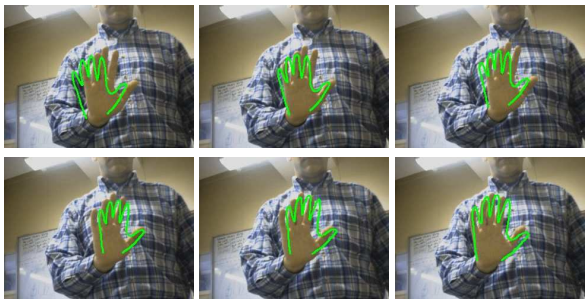


Figure 8: *Tracking example with a cluttered background*

Figure 9: *Example of tracking a head*

3.2 Improvements

Future extensions regarding color cues which aim at making our tracker much more efficient, are discussed hereafter. To our knowledge, color segmentation is not considered in Blake's approaches, in any case for head/hand tracking purpose.

The goal is to achieve a segmentation of regions corresponding to the skin parts in the scene. The requirements comprise a method that can adapt to changing environmental illuminations and complex backgrounds. The segmentation algorithm, which is inspired by [7], consists in two phases:

1. Feature clustering and region growing process based on chromaticity components. This enables the merging process independently of the beginning point and the scanning order of the adjacent regions.

2. Local clustering to refine the segmented regions and a labelling process based on both intensity and chromaticity components to extract skin-colour parts in the observed scene.

Colour can be quantised using different representations, generally called colour spaces. The problem of using colour as a discriminant characteristic for image segmentation raises the question of the best representation for this purpose. The $I_1 I_2 I_3$ space is frequently the one chosen because of its good performance in class separability [8]. Another related question is if the intensity component should be rejected or not. Shin *et al.* [11] show that separability can be significantly affected if the intensity component is neglected. This information is also considered in the second step.

Only the chromaticity components I_2, I_3 are taken into account in the first step of separating skin and non-skin regions. A training phase was performed where the clusters correspond to skin classes as it is shown in figure 10 considering only two components. These classes were interactively learnt beforehand from a large image database.

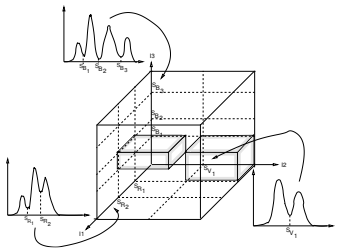
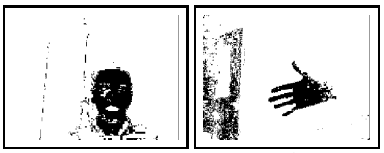


Figure 10: *Classes generation from colour space division*

A first image partition is done on-line by dividing the image in square cells. Potential skin cells included in the colour sub-space defined by the clusters are selected and merged using an adjacency graph.

Then, a second and more selective clustering (figure 10), is achieved on these initial regions by automatically detecting the principal peaks and valleys in the three local I_1, I_2, I_3 histograms [7]. The segmentation results in different regions to be identified. In fact, these regions, sometimes, correspond to skin-colour like entities (figure 11-(b)) in the scene.



(a) (b)

Figure 11: *Examples of colour segmentation: (a) correct segmentation;(b) incorrect identification of skin regions.*

Finally, the means and variances of I_2, I_3 are used to characterise each extracted region and compare with the learnt values in order to identify the skin parts. This permits to filter spurious regions like the one that corresponds to the shelf in figure 3.

Two alternatives are proposed to take into account the colour in the tracking process. First, the segmented (skin labelled) regions can simply delimit interest areas in the image for template fitting, and the approach remains similar to the one described in section 2. Secondly, we can consider the segmented image by replacing the term $\gamma(z_k^i)$ in expression (4) by:

$$\gamma(z_k^i) = \frac{1}{w} \sum_{z_k^i \in I} (1 - l(z_k^i)) + \sum_{z_k^i \in E} l(z_k^i)$$

where I and E are relative to the interior and exterior of the model in image. z_k^i is relative to a pixel and $l(z_k^i)$ is its label – 1 for skin class and 0 otherwise –,and w a weight such as $w > 1$.

4 Conclusion

Tracking methods dedicated to H-R interaction context are supposed to adapt to both changing environmental illuminations and complex backgrounds that may exist in an office or laboratory.

Aiming to track a person or a person's hand through the use of a a video stream, this paper presents the methods used and the obtained results. The results show that it is very difficult to perform tracking using only direct measurements on individual images. It has been shown however that by introducing motion information, captured either using optical flow or background subtraction, the performance of the tracker can be augmented. Results of detection and tracking using these combined criteria are illustrated and demonstrate the validity of the approach. Finally, future extensions are proposed based on colour criteria to improve the robustness.

References

[1] Andrew Blake, Michael Isard, and John MacCormick. *Sequential Monte Carlo Methods in Practice*, chapter Statistical Models of Visual Shape and Motion, pages 339–358. Springer-Verlag, 2001.

[2] G.W. Donohoe, Don R. Hush, and N. Ahmed. Change Detection for Target Detection and Classification in Video Sequences. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 1084–1087, New-York, USA, 1988. IEEE.

[3] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*, chapter An Introduction to Sequential Monte Carlo Methods, pages 3–14. Springer-Verlag, 2001.

[4] I.A Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Based on active Multi-Viewpoint Selection. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 81–87, San Francisco, June 1996.

[5] M.K. Leung and Y.H. Yang. A Region Based Aproach for Human Body Motion Analysis. *Pattern Recognition*, 20(3):321–339, 1987.

[6] Simon Maskell and Neil Gordon. A Tutorial on Particle Filters for on-line Nonlinear/Non-Gaussian Bayesian Tracking. 2001.

[7] R. Murrieta-Cid, M. Briot, and N. Vandapel. Landmarks Identification and Tracking in Natural Environment. In *Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 179–184, 1998.

[8] Yu-Ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Color Information for Region Segmentation. *Computer Graphics and Image Processing*, (13):222–241, 1980.

[9] K. Rohr. Towards Model-based Recognition of Human Movements in Image Sequences. *Computer Vision, Graphics and Image Processing*, 59(1):94–115, January 1994.

[10] S. Rowe and A. Blake. Statistical Feature Modelling for Active Contours. In *European Conf. on Computer Vision*, pages 560–569, 1996.

[11] M. C. Shin, K. I. Chang, and L. V. Tsap. "Does Colorspace Transformation make any Difference on Skin Detection?". In *Workshop on Applications of Computer Vision*, Orlando, FL, 2002.

[12] H.L. Van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley and Sons, Inc., 1968.