Data Fusion for Visual Tracking dedicated to Human-Robot Interaction

L. Brèthes†, F. Lerasle†‡, P. Danès†‡
{*lbrethes,lerasle,danes*}*@laas.fr*
†*LAAS - CNRS*
7 avenue du Colonel Roche, 31077 Toulouse, France
‡*Université Paul Sabatier*
118 route de Narbonne, 31062 Toulouse, France

***Abstract*— The interaction between men and machines has become an important topic for the robotics community as it can generalize the use of robots. In this context, advanced robots must integrate capabilities to interpret humans motion as well as persons gestures in order to perform tasks for the humans or in synergy with them. The purpose of this paper is to show a real-time system for face/hand tracking and hand gesture recognition in the particle filtering framework. We introduce mechanisms for visual data fusion within particle filtering to develop trackers combining in a novel way color and shape cues, skin blobs or frontal face detection. For the purpose of face tracking, the fusion of modalities based on color and shape allows to avoid noticeable drift, even possible subsequent loss in the worst case. For gestures interpretation, an extension is proposed to achieve in the tracking loop the recognition of the current hand posture and of its motion in the video stream. In both tracking scenarios, the combination or fusion of cues proves to be more robust in cluttered environments than any of the cues individually. The global performances of the proposed trackers and future works are also discussed.**

I. Introduction and framework

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. Here, persons are just considered as some "passers by" and no direct interaction is intended. For an active interaction, the robot must also be able to interpret gestures performed by the tracked person. We focus here on communicative gestures to symbolize some referential actions for the robot.

The purpose of this paper is to show a real-time system for face/hand tracking and hand gesture recognition in the particle filtering framework. This formalism has been pioneered in the seminal paper [4] by Isard and Blake. A first reason for focusing on particle filtering as the tracking engine comes from its capability to work with the non-Gaussian noise models required to represent the cluttered environments. A second reason of such a framework is that it allows the information from different measurements sources to be fused in a principled manner. Although this fact has been acknowledged before, it has not been fully exploited within a visual tracking context. Data fusion with particle filters has been mostly confined to skin color and shape cues inside and around simple silhouette shapes [5], while a host of cues (such as motion, color, sound) are sometimes available to increase the reliability of the tracking [11]. In [2], we proposed a preliminary approach based on contours (describing the shape) and skin regions segmentation to track faces and recognize hand configurations in video streams. The integration of skin blobs segmentation on board of our Rackham robot dedicated to H/R interaction (figure 1) showed that its behavior is greatly influenced by the variability of the environment itself (*e.g.* heavy cluttered background) and by the viewing conditions changes in such a mobile robot context. Skin blobs segmentation must be used cautiously and only when necessary.

Fig. 1. Our robot Rackham

Moreover, this cue as well as frontal face detection introduced in [2] are said intermittent because they are unefficient when the person turns back to the camera. This intermittent nature makes them candidate for the design of detection modules, efficient proposal distribution and particle filter initialization as depicted hereafter.

Color distribution on image patches describing the target are proved to be remarkably persistent and robust to changes in pose and illumination [11]. However, this cue remains prone to ambiguity with regard to false alarms characterized by a color distribution similar to that of the region of interest (ROI). These ambiguities can be drastically reduced by introducing shape cues, as human limbs to be tracked are known *a priori* so that silhouette models can be learnt beforehand (figure 5).

The remainder of the paper is organized as follows. Section II briefly outlines the well-known particle filtering formalism and alternative schemes when information from multiple measurement sources are available. Section III presents four cues we aim to combine in trackers dedicated to H/R interaction: frontal face detection, skin blobs detection, contours (shape) and color. Applications of face tracking and gestures recognition (both static and dynamic) are presented in sections IV and V under a variety of conditions/scenarios. Finally, section VI summarises our contribution and opens the discussion for future works.

II. Condensation formalism and data fusion

A. The "Condensation" algorithm

The "Condensation" algorithm —for "Conditional Density Propagation"— is a particle technique for the estima-

tion of the state vector of a nonlinear Markovian system submitted to possibly nonGaussian random inputs [1], [7]. The aim is to recursively estimate the *a posteriori* probability density of the state vector x_k at time k conditioned on the knowledge of past measurements.

Let z_1^k term the available measurements from time 1 to k. At each time k, the probability density function (pdf) $p(x_k|z_1^k)$ is depicted by a set of particles $x_k^{(i)}$ —which are samples of the state vector— affected by weights $w_k^{(i)}$. The idea is to get

$$p(x_k|z_1^k) \approx \sum_i w_k^{(i)} \delta(x_k - x_k^{(i)}), \qquad (1)$$

i.e. to approximate random sampling from the pdf $p(x_k|z_1^k)$ by the selection of a particle with a probability equal to its associated weight. All the difficulty thus lies in the way the particles and their weights are defined all along the estimation process. Moments of the *a posteriori* distribution can then be approximated through the formula $E\big(g(x_k|z_1^k)\big) = \sum_{i=1}^{N_i} w_k^{(i)} g(x_k^{(i)})$.

The estimator initialization consists in the definition of a set of weighted particles which can describe the initial prior $p(x_0)$. Then, starting from a set of weighted particles $\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}$ associated with the filtering density $p(x_{k-1}|z_1^{k-1})$ at time $k-1$, the computation of the particles set associated to the filtering density at next time k proceeds in three steps.

The first step consists in the resampling, within the set $\{x_{k-1}^{(i)}\}$, of N_i new particles $x'^{(i)}_{k-1}$. This resampling is guided by the weight values, in that $P(x'^{(i)}_{k-1} = x_{k-1}^{(i)}) = w_{k-1}^{(i)}$. So, particles associated to high weights $w_{k-1}^{(i)}$ may be duplicated, while low-weighted particles collapse. The consequent uniformly weighted particle set $\{x'^{(i)}_{k-1}, N_i^{-1}\}$ still represents $p(x_{k-1}|z_1^{k-1})$.

Then, each particle $x'^{(i)}_{k-1}$ is propagated between times $k-1$ and k by generating its successor from the pdf $p(x_k|x_{k-1} = x'^{(i)}_{k-1})$ relative to the hidden state vector dynamics. It can be easily shown that $\{x_k^{(i)}, N_i^{-1}\}$ describes the prediction density $p(x_k|z_1^{k-1})$.

Finally, the set of weighted particles associated to the filtering density at time k is determined by taking into account the measurement z_k. The Bayes rule shows that $\{x_k^{(i)}, w_k^{(i)}\}$ describes $p(x_k|z_1^k)$ as soon as each weight $w_k^{(i)}$ relative to $x_k^{(i)}$ is affected the value $p(z_k|x_k = x_k^{(i)})$, prior to a normalization of the $w_k^{(i)}$'s so that $\sum_{i=1}^{N_i} w_k^{(i)} = 1$.

The Figure 2 shows an example. Therein, each ellipse is centered on a particle and has a size related to its weight.

B. Application to visual tracking

Visual tracking can be stated as a filtering problem [7]. The state vector gathers a minimal set of variables relative to the target to be tracked. Its *a priori* dynamics, characterized by $p(x_k|x_{k-1})$, must be consistent with the admissible motions. The target is henceforth parametrized by some position, orientation and size parameters in the current frame. The state vector x_k at time k is made of their values at times k and $k-1$, so as to deal with temporal evolutions of the second order.

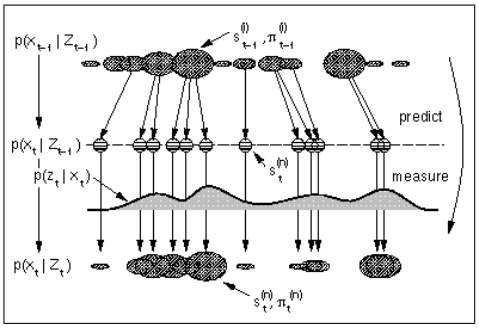


Fig. 2. Outline of "Condensation" algorithm. Blob centers represent particles and blob sizes depict the associated weights (from [7])

A target detection scheme —skin blobs, face detector,...— is applied to the initial frame in order to settle the initial state probability distribution of the tracker.

Several choices of the likelihood $p(z_k|x_k^{(i)})$ can be considered. It will be further assumed that when M measurement sources are used, they are independent conditioned on the knowledge of the state vector, so that $p(((z_1), \ldots, (z_M))_k|x_k^{(i)}) = \prod_{m=1}^{M} p((z_m)_k|x_k^{(i)})$.

C. Enhanced schemes

This paper also considers two enhancements of the original Condensation scheme: the ICondensation algorithm will be used for face tracking (section IV), while the Mixed-State Condensation will handle multiple gestures recognition (section V).

1) ICondensation: A deeper insight to the Condensation algorithm shows that the resampling step is necessary to avoid that after a few iterations, all but one particle have negligible weights. Indeed, such a degeneracy phenomenon cannot be avoided whatever the recursive particle filtering strategy. Yet, another fact can be used to limit this problem in addition to resampling, namely the choice of the importance —or proposal— density, *i.e.* of the way the particles are distributed in the state space [1].

Positioning the particles according to the stochastic state dynamics, as is the case in Condensation, isn't the optimal choice. Instead, the successor at time k of a particle $x_{k-1}^{(i)}$ should be drawn from an importance density $\pi(x_k|x_{k-1}^{(i)}, z_k)$ combining both the dynamics $p(x_k|x_{k-1}^{(i)})$ and the actual measurement z_k [3]. Notice that a systematic procedure is defined so as to update its weight accordingly.

The ICondensation algorithm [5] is a step towards this aim. In this approach, particles at time k can be drawn from a pdf of the form $\pi(x_k|z_k)$ according to the ROIs in the current frame. Practically, they can be selected in the vicinity of color blobs (section III-B).

However, if a particle drawn exclusively from the image data is inconsistent with its predecessor from the point of view of the state dynamics, the update formula leads to a small weight. In order to avoid this problem, the ICondensation implementation also draws some particles

following the original Condensation scheme and others using the prior distribution $p(x_0)$.

2) A Condensation algorithm for jump Markov systems: The Condensation algorithm can also be readily extended to tackle jump Markov systems, see the "mixed-state" version of [6].

Let l_k be a variable taking its values in a discrete set —typically a gesture index— and following a discrete-time Markov chain with known transitions probabilities T_{ij}. Assume that the state vector x_k and the measurement z_k obey a known jump Markov system such that

$$\begin{aligned} &p(x_k|x_{k-1}, l_k, l_{k-1}) = p(x_k|x_{k-1}, l_k) = p_{l_k}(x_k|x_{k-1}) \\ &\text{and } p(z_k|x_k, l_k, l_{k-1}) = p(z_k|x_k, l_k) = p_{l_k}(z_k|x_k). \end{aligned} \quad (2)$$

Stating $X_k = (l_k, x_k)$ enables to deal with such a system in the Condensation framework.

On the one hand, in the prediction step a particle $X_k^{(i)} = (l_k^{(i)}, x_k^{(i)})$ is sampled from the dynamics prior

$$\begin{aligned} p(X_k|X_{k-1}^{(i)}) &= p(l_k, x_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}) \\ &= p(x_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}, l_k)p(l_k|l_{k-1}^{(i)}, x_{k-1}^{(i)}), \\ &= p_{l_k}(x_k|x_{k-1}^{(i)})T_{l_{k-1}^{(i)} l_k}. \end{aligned} \quad (3)$$

A straight way to perform this is to first sample the discrete index $l_k^{(i)}$ from the transition probabilities $T_{l_{k-1}^{(i)} l_k}$, and then draw $x_k^{(i)}$ from $p_{l_k^{(i)}}(x_k|x_{k-1}^{(i)})$.

On the other hand, the measurement update step involves the likelihood $p(z_k|X_k^{(i)})$.

A MAP estimate $\hat{l}_k$ at time k can be deduced, based on the sum of the weights of all the particles having the same discrete index at this time, and an estimate $\hat{x}_k$ follows:

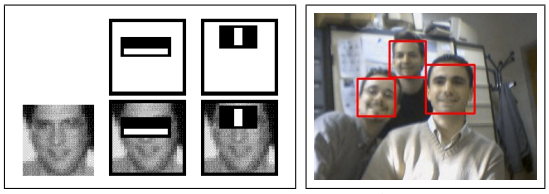
$$\begin{aligned} \hat{l}_k &= \arg\max_l \textstyle\sum_{i \in \Upsilon_l} w_k^{(i)}, \text{ with } \Upsilon_l = \{i : X_k^{(i)} = (l, x_k^{(i)})\} \\ \hat{x}_k &= \frac{\sum_{i \in \hat{\Upsilon}} w_k^{(i)} x_k^{(i)}}{\sum_{i \in \hat{\Upsilon}} w_k^{(i)}}, \text{ with } \hat{\Upsilon} = \{i : X_k^{(i)} = (\hat{l}_k, x_k^{(i)})\}. \end{aligned} \quad (4)$$

III. Measurement cues

We first focus on intermittent cues such as frontal face and skin regions detection. We then deal with persistent cues such as color and shape.

A. Frontal face detection

The method used for face detection was introduced by Viola *et al.* [12]. It is based on a boosted cascade of Haar-like features. These features are obtained by substracting the sum of the pixels lying inside the white rectangles from the sum of the pixels in the dark rectangles (Figure 3(a)). They enable the detection of relative darkness between



(a) (b)

Fig. 3. (a) Haar-like features overlaying on a training face, (b) example of face detection

eyes and nose/cheek or nose bridge. An over complete set of features is generated by scaling the Haar-like masks independently in vertical and horizontal directions. A cascade of classifiers is a degenerated decision tree where at each stage a classifier is trained to detect almost all frontal faces while rejecting a certain fraction of non-face patterns. This way, background regions are quickly discarded while focusing on promising frontal face-like regions (figure 3(b)).

B. Skin regions detection

Human skin colors have a specific distribution in color space. They can be clustered to form a feature space for segmentation. A color histogram model learnt offline is classically used to classify skin-like pixels. In our approach [2], a watershed-based segmentation is then applied on the labeled pixels to segment the skin regions.

Several color segmentation techniques have been used for skin blobs segmentation [8] using a skin pixel classification. However, in a mobile robot context, these are generally influenced by the variability of the environment clutters and the associated viewing conditions changes. Typically, overexposure (when the robot is close to a bay window) or underexposure (when the robot moves in a corridor) make more uncertain the separation of skin regions from background. Moreover, for cluttered environments, spurious close-to-skin colored regions can be sometimes segmented, for example wooden doors and desks (figure 4(b)). Yet, part of such false alarms can be eliminated regarding the aspect ratio of the region. Clearly, skin detection must be used cautiously and combined with other cues. Figure 4 shows two examples of correct and incorrect skin region segmentations.

(a) (b)

Fig. 4. Two examples: (a) correct segmentation, (b) incorrect segmentation due to clutter in background

C. Shape cue

The use of shape cue requires that the class of targets to be tracked is known *a priori* and that sufficiently precise silhouette models can be learned beforehand. Such conditions are met in human limbs tracking applications where coarse shape cues (of head or hand) can be used.

The aim here is to track faces and well-defined hand postures that represent a limited set of commands that the users can give to the robot. To use a simple view-based shape representation, face and hand are therefore represented by coarse 2D rigid models, *e.g.* their silhouette

contours, by means of splines [7]. These models, although simplistic, permit to reduce the complexity of the involved computations and remain discriminatory enough to track a set of known hand postures in complex scenes as will be shown later. Examples of these models are presented in figure 5.

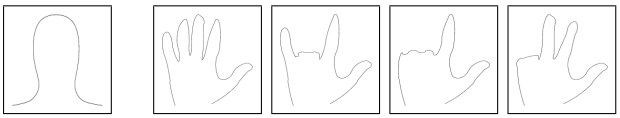


Fig. 5. Templates for face or hand configurations (depending on the number of open fingers)

In the particle filter measurement update step, each sample is classically given a likelihood that depends on the sum of the squared distances between model points and corresponding image points [7]. The model points are chosen to be uniformly distributed along the spline. The corresponding ones are found by searching in the image for color edge points which lie on the spline normals that pass through these points (figure 6).

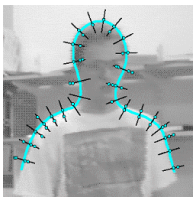


Fig. 6. Measurement model for shape cue (from [7])

Color gradient can be estimated in various ways. It can be computed either as a combination of gradients issued from each channel separately, or as a vector to make full use of color information. We follow this last principled way (see [2] for more details).

Unfortunately, for cluttered background, using only shape cue for template fitting is not sufficient, as irrelevant contours may attach the tracker. Moreover, due to unfavourable illumination, the target contour may not be as prominent as expected, while color cues are known to be more robust to such lighting conditions.

D. Color distribution on image patches

1) Basics: For the color distribution modeling, we use independent normalized histograms computed in the RGB space. We denote the B-bin reference histogram model in channel $c \in \{R, G, B\}$ by $h^c_{ref} = (h^c_{1,ref}, \ldots, h^c_{B,ref})$. The color distribution $h^c_x = \left(h^c_{1,x}, \ldots, h^c_{B,x}\right)$ of a region B_x corresponding to any state x is computed by

$$h^c_{j,x} = c_H \sum_{u \in B_x} \delta_j(b^c_u), j = 1, \ldots, B, \qquad (5)$$

where $b^c_u \in \{1, \ldots, B\}$ denotes the histogram bin index associated with the intensity at pixel u in channel c of the color image z^C, δ_a terms the Kronecker delta function at a, and c_H is a normalization factor ensuring that $\sum_{j=1}^{B} h^c_{j,x} = 1$.

The color likelihood model must be defined so as to favor candidate color histograms close to the reference histogram. A popular measure between two distributions $h_1 = \{h_{j,1}\}_{j=1,\ldots,B}$ and $h_2 = \{h_{j,2}\}_{j=1,\ldots,B}$ is the Bhattacharyya coefficient [9]:

$$D(h_1, h_2) = (1 - \sum_{j=1}^{B} \sqrt{h_{j,1}.h_{j,2}})^{1/2}$$

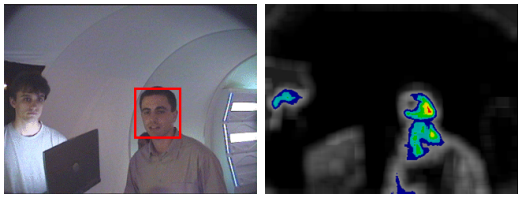
The smaller D is, the more similar the distributions are. Finally, the likelihood of a state x when faced to z^C is given by

$$p(z^C|x) \propto \exp(- \sum_{c \in \{R,G,B\}} D^2(h^c_x, h^c_{ref})/2\sigma^2_C).$$

If the tracked region contains different patches of distinct colors, e.g. the face and clothes of a person, the histogram-based modeling will capture them. It suffices to split the ROI into subregions, each with its own reference color model [10]. We consider the partition $B_x = \bigcup_{p=1}^{N_R} B_{p,x}$ associated with the set of reference histograms $\{h^c_{p,ref} : c \in \{R, G, B\}, p = 1, \ldots, N_R\}$. By assuming conditional independence of the color measurements within the different subregions defined by the state x, the multi-region color likelihood becomes:

$$p(z^C|x) \propto \exp(- \sum_{c \in \{R,G,B\}} \sum_{p=1}^{N_R} D^2(h^c_{p,x}, h^c_{p,ref})/2.\sigma^2_C)$$

where the histogram $h_{p,x}$ is collected in the region $B_{p,x}$. The histogram based definition of the color likelihood is illustrated in figure 7.



(a) (b)

Fig. 7. (a) ROI, (b) color likelihood of the location only (scale factor fixed)

2) Model update: Illumination conditions, out-of-plane rotated faces, visual angle as well as camera parameters are known to be difficult to handle as they may lead to an inaccurate tracking or a complete loss of lock. To overcome these appearance changes, we update the target model during slowly changing image observations. This update is made according to

$$h_{ref,k} = (1 - \alpha).h_{ref,k-1} + \alpha h_{E[x_k]}$$

where k terms the frame time and α weights the contribution of the mean state histogram $h_{E[x_k]}$ *w.r.t* the target model $h_{ref,k-1}$. The contribution of a specific frame decreases exponentially in time. The channel index c and bin index j have been omitted for compactness reasons.

IV. Application to face tracking

Color-based filtering schemes enable the robust tracking of targets undergoing complex changes in shape and appearance. Unfortunately, due to the model updating, noticeable drifts or even loss of target can be observed [11]. The robustness of the tracker to drifts and color clutters can however be increased by incorporating multi-patches color models and by fusing color and shape cues in the measurement model. This is illustrated in snapshots from a sequence of 300 frames (figure 8).

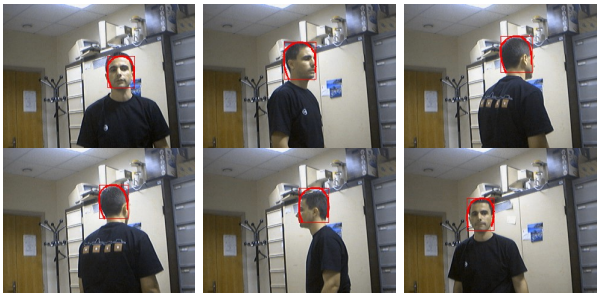


Fig. 8. Face tracker fusing color and shape cues (images $58, 129, 142, 159, 188, 234$): better postioning on the target, weak drift

Moreover, considering multi-patches (on face and especially clothes) of distinct color distribution makes the tracker keep focusing on the current target even if several persons enter in the view field of the camera (figure 9).

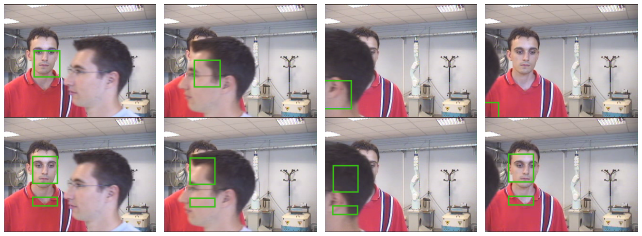


Fig. 9. Influence of the multi-part color model in the tracker images $246, 254, 261, 268$): (top) with a single part, the tracker locks to a wrong target in the foreground. (bottom) with multi-part model, the tracker keeps locked onto the ROI even after an occlusion

As for the computational cost, a non-optimized implementation tracks regions of about 20×20 pixels at a rate of 50 fps with $M = 200$ particles on a $3GHz$ Pentium IV.

Last improvements concern the tracking of targets undergoing rapid motion, persistent occlusion or (re)-apparition in the scene. The aim is to make the tracker robust to such events, which generally would cause target loss.

In the Condensation scheme, the state evolution model is used to sweep the image ROIs in which the target is predicted to lie. This precludes any re-initialization of the tracker when the target can be anywhere in the image. The ICondensation depicted in section II fulfills such problems by using proposal distributions based on face detection or skin blobs detection (low-level) and by fusing color and shape cues in the likelihood model (high level).

Let B be the number of detected blobs. Following [5], the centroid of each blob is computed as a coordinate b'_i in the original image, and a 2D importance function π is defined by the Gaussian mixture

$$\pi(x_k|z_k) = \sum_{i=1}^{B} \delta_i . \mathcal{N}(b_i, \Sigma_B)$$

where $b_i = b'_i + \bar{x_B}$, and $\bar{x_B}$ and Σ_B are the mean and covariance respectively of the offset from the blob (or face) position to the centroid of the contour describing the face. These parameters are learned offline by using a contour tracker and comparing the output of skin blobs (or face) detection with the centroid of the tracked contour.

V. Application to gestures recognition

In [2], we proposed a preliminary approach to the recognition of the current hand posture (figure 5) and the automatic switching between multiple templates in the tracking loop. For a richer interaction, an extension of this tracker is proposed so as to handle multiple canonical motion models as classifiers for gesture recognition.

The bayesian mixed-state framework depicted in section II is well-suited to manage hand motion and configuration models in video streams. Indeed, it suffices to augment the state vector by two discrete variables respectively indexing the configuration and the motion type. So far, these indexes have been assumed mutually independent, and evolve over time according to distinct transition probabilities matrices. A further step will consist in defining these switching probabilities from the predefined interaction language.

As aforementioned, color and shape modalities are mixed in the measurement model while the extension to multi-part color modelling is efficient to discriminate between configurations. This last issue is achieved within our color model by splitting the tracked region into subregions corresponding to the palm and fingers, and by considering a single reference color model which is related to the palm in the previous frame. Local Bhattacharyya distances on these ROIs can exhibit the presence or absence of open fingers, thus improving the discriminative power between templates associated to configurations. Pratically, the smaller the color discrepancy between a given ROI and the reference model, the higher is the probability that an open finger is located inside this ROI.

Regarding the experimentations, we consider two main scenarios. In the first one, the behavior of the tracker is illustrated when using only color cue. In the second one, we fused color and shape cues to recognize both hand postures and motion models.

A. Considering color cue only

With no assumption regarding hand silhouette templates, that is, considering only color cue, a moving hand can be tracked with a reasonable accuracy as shown in the sequence of figure 10. In this sequence, the contour is drawn in green (resp. red) during roughly horizontal (resp.

vertical) motions and in blue if the hand remains stationary. The classification of motion by model-switching is accurate in most cases and the tracker runs at about $60\ Hz$.

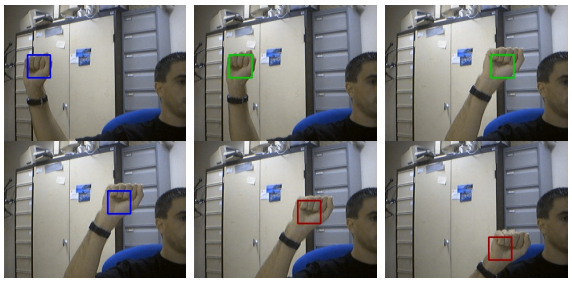


Fig. 10. Hand tracker based on only color distribution: images $29, 32, 46, 47, 56, 64$

B. Fusing color and shape cues

Figure 11 shows a few snapshots from a sequence of 200 frames, where the hand moves in front of a cluttered background while its posture and global motion changes. In this sequence, the contour is drawn in pink (resp. green) during roughly horizontal (resp. vertical) motions and in blue if the hand remains stationary.

Fig. 11. Fusing color distribution and edges in the hand tracker: images $26, 32, 44, 45, 58, 71, 93, 104, 152$

Recognition results are compared with a ground truth for both hand postures and motion models. While the hand posture is correctly determined in most frames (close to 99%), the motion model is more often misclassified: about 75% for vertical or horizontal models, only 65% for the stationarity (due to hand shivering).

VI. Conclusion and future works

In this paper we introduced mechanisms for data fusion within particle filtering to develop trackers combining color, edges based cues, eventually skin blobs or frontal face detection in a novel way. Being the most persistent, the two first cues were used as the mains cues for tracking. The two last ones, logically intermittent, act in detection and initialization modules for the particle filter.

For face tracking purpose, the fusion of color distribution and edges-based modalities allows to avoid noticeable drift and possible subsequent loss, experienced sometimes by considering these cues individually. Considering multiple subjects in the view field, multi patches of distinct color distribution (such as the face and clothes) allows the tracker to keep focusing on the current target. For gestures tracking/recognition purpose, our tracker was adapted to track multiple templates (representing hand postures) and associated motion models. In both tracking scenarios, the combinaison or fusion of cues proved to be more robust than any of the cues individually. Videos of the diffrent trackers are available at www.laas.fr/∼lbrethes/icra05/

Furthermore, we want to involve the fusion of other information such as sound or motion intermittent cues (which are less prone to clutter) and adapt our tracker to be able to track multiple targets simultaneously. The multiple target tracking could also be applied to the two-handed gestures which is of great interest.

VII. Acknowledgements

The work described in this paper was partially conducted within the EU Integrated Project COGNIRON (”The Cognitive Companion”) and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

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

[2] L. Brèthes, P. Menezes, F.Lerasle, and M. Briot. Face Tracking and Hand Gesture Recognition for Human-Robot Interaction. In *Int. Conf. on Robotics and Automation (ICRA'04)*, pages 1901–1906, 2004.

[3] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Cambridge University Department of Engineering, 1998.

[4] M.A. Isard and A. Blake. CONDENSATION-Conditional Density Propagation for Visual Tracking. *Int. Journal of Compution Vision (IJCV'98)*, 29(1):5–28, 1998.

[5] M.A. Isard and A. Blake. Icondensation: Unifying low-level and high-level Tracking in a Stochastic Framework. In *European Conf. on Computer Vision (ECCV'98)*, pages 893–908, 1998.

[6] M.A. Isard and A. Blake. A Mixed-state Condensation Tracker with Automatic Model-switching. In *Int. Conf. on Computer Vision (ICCV'98)*, pages 107–112, Bombay, 1998.

[7] M.A. Isard and A. Blake. Visual Tracking by Stochastic Propagation of Conditional Density. In *European Conf. on Computer Vision (ECCV'96)*, pages 343–356, Cambridge, April 1996.

[8] M.J. Jones and J.M. Rehg. Statistical Color Models with Application to Skin Detection. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, pages 274–280, 1999.

[9] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptative color-based particle filter. *Journal of Image and Vision Computing*, 21:90–110, 2003.

[10] P. Pérez, J. Vermaak C. Hue, and M. Gangnet. Color-based probabilistic tracking. In *European Conf. on Computer Vision (ECCV'98)*, pages 661–675, 2002.

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

[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.