# Tracking using Particle and Kalman Filters in Hand Washing Quality Assessment System

I. Parra, D. Fernández, M.A. Sotelo, M. Marrón, M. Gavilán, G. Lacey

***Abstract*** **– This paper describes a method for tracking the hands/arms of a person performing hand washing. A hand washing quality assessment system needs to know if the hands are joined or separated, if they are under water, if they are in contact with the towel or the tap, and it has to be robust to different lighting conditions, occlusions, reflections and changes in color on the steel surface. In the proposed system hands/arms are extracted by using skin color segmentation. An area based ellipse model is used for representing each hand/arm. A Particle filter (PF) in combination with a k-means based clustering technique is used for tracking both hands/arms. A supervision algorithm measures the number of objects being tracked and the quality of the tracking itself. Finally the PF performance is discussed and compared with the standard Kalman filter (KF) estimator.**

***Keywords*** **– Hand washing, Kalman filter, Particle filter, Skin detection, Tracking.**

## I. Introduction

Hand washing is a critical activity in preventing the spread of infection in health-care environments. Several guidelines recommended a hand washing protocol consisting of six steps that ensure that all areas of the hands are thoroughly cleaned [1]. A multi-class SVM classification of the hand gestures by using HOG descriptors is presented in [2] to monitor the user's hands motion in order to ensure that the hand washing guidelines are correctly followed. The system observes the user with a camera mounted above the sink and it needs to know if the hands are joined or separated, if they are under water, if they are in contact with the towel or the tap, etc. Accordingly an object tracking method becomes mandatory for a robust hand/arm pose estimation.

In [3] the hands and the towel are modeled as a *flock* of features describing its approximate shape and three independent particle filters, one for each of the right and left hands, and one for the towel, are used. In [4] a single Extended Particle filter (XPF) is used to track multiple and dynamic objects in complex environments where a multi-modal distribution represents the most probable estimation for each object position and velocity.

In our approach a skin color segmentation process is applied. The hands/arms are modeled by using an area based ellipse fitting method. A single multi-modal distribution is then used in order to estimate the position and orientation of both hands/arms. That is not the case of the KF which needs two different filters for each one of the hands. The results obtained by PF are analyzed and compared with those given by KF estimator.

The remainder of the paper is organized as follows: Section II provides a description of the hands/arms segmentation process and the proposed model. PF is described in Section III. The results achieved up to date are presented in Section IV. Finally, conclusions and the description of our future lines of research are presented in Section V.

## II. Hands/arms motion model and measure equation

### A. The model equations

The main objective of the proposed method is to model the hands/arms movements of a person washing their hands. In order to achieve this goal each one of the hands/arms has been characterized as an ellipse (Figure 1) represented by the following state vector:

$$\vec{x} = \{x_k, y_k, \theta_k, \dot{x}_k, \dot{y}_k\} \tag{1}$$

where $x_k$, $y_k$, $\dot{x}_k$, $\dot{y}_k$, are the ellipse position and velocity and $\theta_k$ is its orientation. The ellipse axis were not included in the state vector because they added complexity to the measurement estimation without any improvement and it required a higher computation time. Accordingly their values have been fixed taking into account the distance from the camera to the

I. Parra, D. Fernández, M.A. Sotelo, M. Marrón and M. Gavilán are with the Electronics Dept. University of Alcalá, Madrid. Spain **parra,llorca,sotelo@depeca.uah.es**

G. Lacey is with the Computer Science Dept. Trinity College Dublin. Ireland **gerard.lacey@cs.tcd.ie**

1-4244-0830-X/07/$20.00 ©2007 IEEE.

washbasin. With these states, the discretized system dynamics are given by:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \vec{w}_k \qquad (2)$$

where $T$ is the sampling period and $\vec{w}_k$ is the noise vector related to the system which determines the spread capability of the particles that identify each hand.

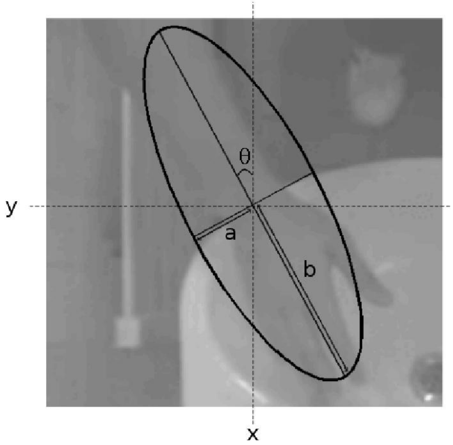


Fig. 1. PARAMETERS OF THE ELLIPSE MODEL

It is assumed that measurements $z_t$ are mutually independent and also with respect to the dynamic process. To represent the measurement process a simple model is used in which Gaussian noise is added. The measurement vector is defined by the ellipse position $(x_k, y_k)$ and orientation $(\theta_k)$:

$$\vec{y} = \{x_k, y_k, \theta_k\} \qquad (3)$$

and the measurement equation is then given by:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \vec{v}_k \qquad (4)$$

where $\vec{v}_k$ is the noise vector related to the accuracy of the measurements.

### B. Adapting vision measurements to the proposed model

An area-based ellipse fitting over a skin probability map is used to estimate the hands/arms position and orientation. A skin color segmentation method is used to generate the skin probability map. The aim is to have a probability map with high intensity values in the skin pixels and low intensity values in the non-skin pixels. Skin detection plays an important role in various applications such as face detection, searching and filtering image content on the web, video segmentation, face/head tracking, etc. Among the different color spaces we use the normalized RGB color space, which is easily obtained from the RGB values by a simple normalization procedure:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \qquad (5)$$

The normalization removes intensity information, so that $rgb$ values are pure colors. Because $r + g + b = 1$ no information is lost if only two elements $(r, g)$ are considered. In that case the color space is usually named as *rg-chromaticity*. In the work described in [5] the illumination influence over the skin-tone color for several nationalities is done by using four fluorescent lamps with different CCTs. Thus trapezoidal areas in the rg-chromaticity plane group the different skin color values of the different subjects nationalities according to the illumination conditions. In order to have a more robust appearance of the skin-tone color with different lighting conditions a lighting compensation or color constancy step is applied. The grey-world algorithm [6] modified in [7] is used. Then the skin segmentation is applied by using a rectangular area in the rg-chromaticity plane. Four boundaries are defined $(r_{min}, r_{max}, g_{min}$ and $g_{max})$ and the skin probability is then modeled by a Gaussian function:

$$f(r,g) = \frac{1}{2\pi\sigma_r\sigma_g} \exp\left(-\frac{(r-r_{mean})^2}{2\sigma_r^2} - \frac{(g-g_{mean})^2}{2\sigma_g^2}\right) \qquad (6)$$

where $r_{mean}, g_{mean}, \sigma_r$ and $\sigma_g$ are the mean and the variance values for each chromaticity channel respectively. Variances are fixed to a practical value of $0.6$ and $r_{mean}$ and $g_{mean}$ are computed according to the rectangular boundaries: $r_{mean} = (r_{max-} - r_{min})/2$ and $g_{mean} = (g_{max-} - g_{min})/2$.

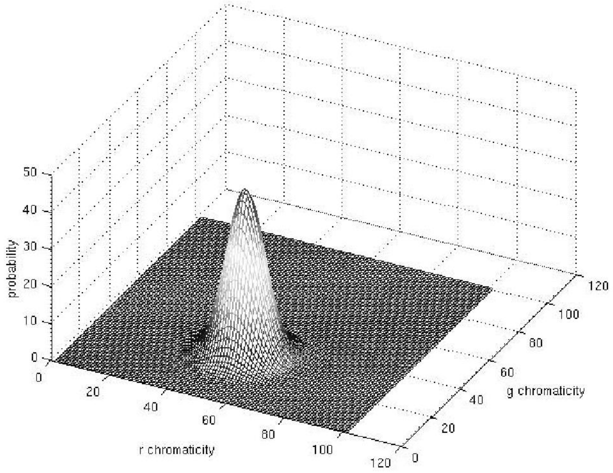


Fig. 2. GAUSSIAN DISTRIBUTION IN THE RG-CHROMATICITY PLANE

If the chromaticity of a pixel falls in the modeled area (see Figure 2), then its probability is computed by using equation (6) and normalized between 0 and 255. Thus the intensity of the result images represents the probability of a pixel of being skin. Figure 3 depicts two examples of segmented images.

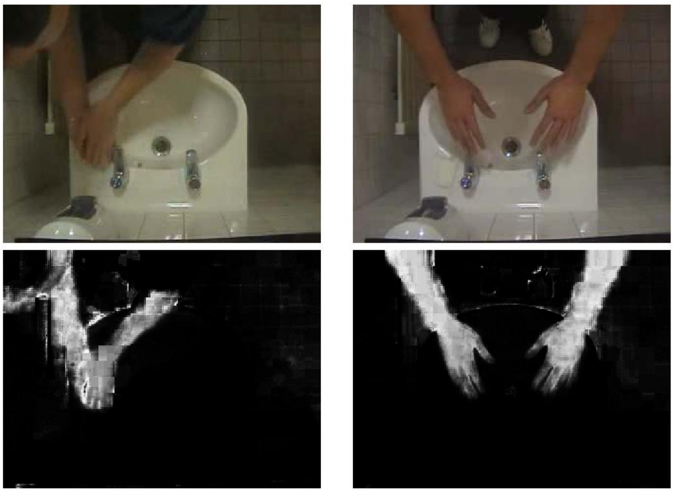


Fig. 3. UPPER ROW: ORIGINAL IMAGES. LOWER ROW: SKIN SEGMENTATION IMAGES

## III. PARTICLE FILTERING METHOD

Algorithms using PF to track one or several objects were named as *Condensation* [8]. The motivation for these filters is to solve the problem of tracking a variable number of objects, or tracking in cluttered environments, or when the models used to represent the objects are the same for each one of them. In the last situation using independent filters to track each one of the objects is not the best solution from the computational point of view and also because independent filters tend to join over the same target. In addition KF is not optimal because it is based on uni-modal Gaussian densities and it is not able to represent multiple alternative hypothesis. Although PF usually are more time consuming, they are also independent on the number of objects being tracked [9], [10].

In this application a uni-modal representation of the system would imply to manage a $10 \times 1$ state vector and the number of particles needed to model all the possible states will become intractable. Instead of that, the estimation for each one of the arms is merged in a single Gaussian multi-modal probability density function (pdf) which integrates the state information for each one of the hands/arms. This poses two main challenges. First, to avoid that one of the hands/arms absorbs all the particles of the filter, leaving a "uni-modal" filter. In a multi-tracking system where the models used to represent the objects are the same for all of them there is always one object which yields a higher probability than the rest. In our system the number of objects to track is known, and this is used to propagate the particles in an oriented way, assuring the correct representation of the multi-modal pdf. The prior knowledge of the state pdf allows us to force the particles to fill the states we know will represent the system real state. This is performed with a k-means clustering technique which keeps the system from merging in a single Gaussian pdf or to spread in more than two which should represent the two arms. The second challenge is to make the tracking robust to partial occlusions at the time it follows quick changes of the targets. This implies a trade-off solution between a fast response and a stable estimation to partial occlusions. To face this a supervision algorithm which takes care of the consistency of the estimations has also been developed. This supervision algorithm controls the filter estimations for each one of the hands/arms at every filter iteration, and adjusts the filter parameters depending on the occlusion, position and velocity of the arms and the quality of the estimations. Even if the filter degrades it is restarted. This supervision algorithm allows for stable estimations as well as fast filter responses.

The filter can be divided in the following sub-systems (Figure 4):

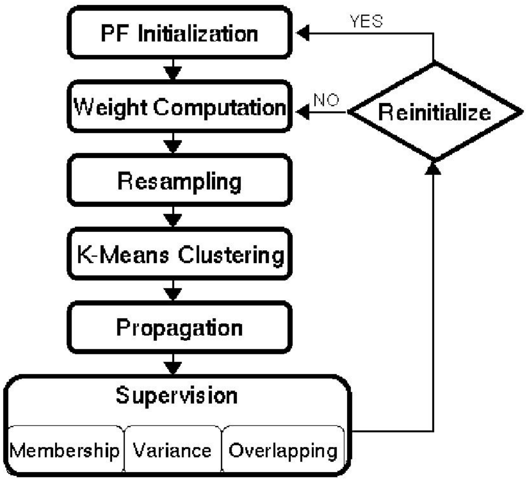


Fig. 4. OUTLINE OF THE PARTICLE FILTER ESTIMATOR

### A. *Initialization*

In order to cover as much as possible of the probability density function of the state a fixed number of randomly generated particles are created. Also every time the supervision system decides that the estimation of the state has not enough quality the filter will be restarted and new randomly generated particles will take the place of the old ones.

### B. *Weight computation*

At this step two tasks are carried out. First each particle is scored with the probability of being the real state (position, velocity and orientation) of the arms. Second the estimated state for each one of the arms is computed.

The probability of representing the real state of the system is computed, for each particle, using the area-based ellipse fitting method explained in Section II B. For each particle, its weight is computed by summing up the membership to the skin function of all the ellipse inner points (Equation (7)).

$$Membership(X) = \sum_{p_x,p_y \in ellipse} f(p_x,p_y) \qquad (7)$$

where $f(p_x, p_y)$ is the probability of the point $(p_x, p_y)$ of being a skin pixel, yielded by equation (6).

This way as long as a particle/ellipse cover image regions with high probability of being skin it will get high scores. Particles with high scores will have higher probability of being regenerated in the re-sampling step.

Once the weight is computed for all the particles the estimation of the real state of the system is also calculated. Different techniques has been explored to find out the one that best represents the real hands/arms position and velocity as well as allowed for a fast filter response to fast movements and partial occlusions. Conservative techniques which use all the filter particles in the estimation of the system state yielded not accurate estimations because the filter dynamics and observation models have been adjusted to quickly respond to changes in the system state. This make the filter always have an important number of particles at states "surrounding" the real one which will quickly absorb fast changes (movements of the arms,occlusions,..). But these particles also corrupt the measure. To get an accurate estimation of the state these particles have to be removed from the computation of the estimated state. To do so the estimated state is computed using a weighted average for an elite of the particles. Using only particles weighted 80% over the average membership of the particles, the real state is computed for each one of the hands/arms clusters as follows:

$$\hat{X}_i = \sum_{X_j/Membership(X_j)>0.8*Memb_{mean}} X_j \qquad (8)$$

It has to be noticed that by this time the particles already has been labeled as belonging to one of the arms and so (8) is computed separately for the two hands/arms using only its particles. At the first filter iteration the particles are randomly assigned to one of the arms and in the next iterations the clustering algorithm will label them (see D. Clustering).

This, along with the supervision algorithm, allows our system to be soft and accurate in the tracking as well as quick to changes in the state of the system. When this elite is not used the estimated state degrades quickly.

### C. Re-sampling

In the re-sampling step new particles are randomly regenerated from the old ones depending on the weight/score received on the previous step. Particles receiving a higher score are more likely to be regenerated. To do so if we have $N$ particles, a segment $[0, 1]$ is divided into $N$ sub-segments $[na1, na2]$ whose length is proportional to the score got by each one of the particles. Then a random number is generated $(rand([0\ 1]))$ and the sub-segment in which the random number is inside will determine the particle to be regenerated as the particle which created this sub-segment.

In the re-sampling process a counter keeps a record of the number of particles regenerated for each cluster/arm and forces the re-sampling to regenerate exactly the same number of particles for each one of the clusters. If the re-sampling regenerates a particle belonging to an arm which has already reached the maximum number of particles, this regeneration is ignored and a new re-sampling step is performed. In our system particles are equally distributed between the two arms.

### D. Clustering

A k-means clustering algorithm is used to split the particles into two clusters (the two arms) after the re-sampling. This information will be used in the next weight computation and re-sampling steps.

The k-means clustering algorithm aim is to place a set of vectors in the input space which describe in a discrete way the density of observed samples. To do so it places K random seeds and take them as centroids, linking the samples to their nearest centroid. The samples assigned to each centroid will make a cluster. The centroids are then recalculated as the mean of all of its samples. The algorithm is then repeated until the centroids variation is under a certain threshold. The k-means algorithm uses $(x, y)$ particles position in the image as input. The output is a list of labeled particles as 1 or 0. Once the particles have been labeled we have to link the new clusters to the last arms estimation. A simple minimum distance criterion is used.

### E. Propagation

Applying the dynamic model of the system to the particles these are propagated to the next state. This propagation step is tuned up by the supervision algorithm at every filter iteration just modifying the variance of the noise added to the dynamic model. There are three different situations which will be handled by the supervision algorithm:

- Separated arms, fast movements: when the arms have no overlapping and last movements have been fast, extra noise is added to the dynamic model so that the predictions adjust better the real movements of the arms.
- Joined arms, fast movements: when the arms have some overlapping and movements have been fast no extra noise is needed.
- Joined arms, great overlap: when the arms have great overlapping and they are very close to each other a decrease in the noise of the dynamic model is needed in order to avoid the joining of the two arms and also to maintain good estimations to partial occlusions.

This tuned up propagation step predicts the evolution of the system as a whole taking into account not only the last targets position and velocity but also the system most probable evolution in the next iterations. With this tuning step our filter is able to follow fast changes in the state as well as to keep good estimations to partial occlusions.

### F. Supervision

The supervision algorithm controls the PF state and restarts it when its estimation has degraded. The main reasons for a degradation in the estimations are occlusions, bad initial estimations or failures in the skin extraction method. The supervision algorithm uses five indicators to evaluate the quality of the filter estimation:

- Membership of the state estimation: For each iteration the membership of the state estimated by the filter is computed and asked to reach a minimum level. If it does not reach this minimum membership for several consecutive iterations the supervision algorithm decides that the tracking has lost one or both arms and restart the filter.
- Variance of the clusters position with respect to the estimated state: For each one of the clusters the variance of its particles is computed as follows:

$$var_i = \frac{1}{N} \sum_{x_k/x_k \in cluster_i} \sqrt{dist(x_k, x_{opt})} \qquad (9)$$

  where $x_{opt}$ is the estimated state. If the quality of the estimations decreases then variance becomes greater because the filter can not set particles around an estimation with high probability. Then the supervision algorithms check for this situation in several iterations and if it remains restarts the filter.
- Overlapping: To prevent the estimations from joining in one single pdf, the overlapping of the solutions for the arms position is computed on every iteration. If this value is above a threshold the supervision algorithm restarts the filter.

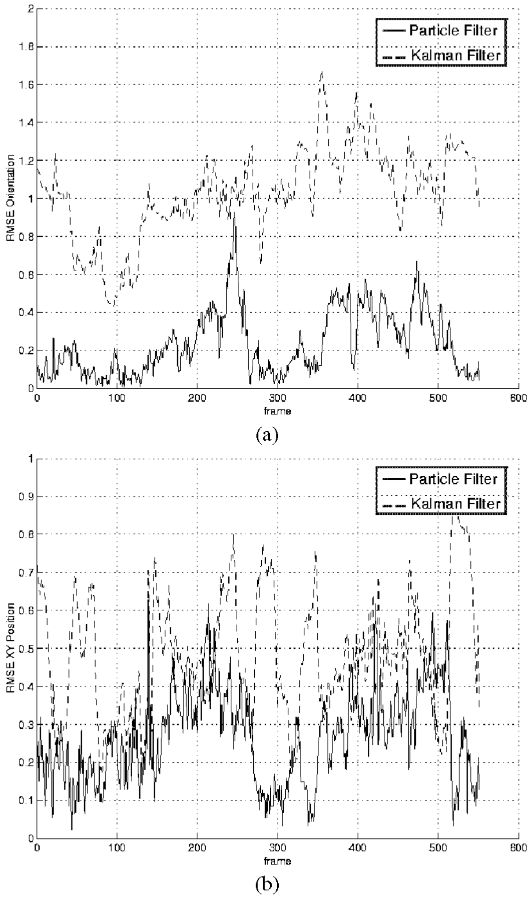
Using this information the supervision algorithm estimates the number of object being tracked, the tracking quality and also is able to determine if both clusters have been assigned to the same arm or not. In addition the dynamic model is tuned up depending on the above parameters. When the arms are far from each other and with no overlapping between them the filter tends to predict faster and longer movements. When the arms are close to each other and overlapping is big the filter is conservative to keep the particles from swapping from the arm with the lower score to the one with the higher. If the last happened the arm with the higher score will begin to steal particles from the other arm leaving this one without particles in approximately a hundred of iterations. Then the supervision algorithm will detect the overlapping and will restart the filter.

## IV. Results

In order to evaluate the performance of both, the Particle and Kalman filters, a set of videos (about 600 frames each, 24 seconds long) have been recorded. Each video consists of a sequence with different subjects washing their hands, with different light conditions and some of them wearing wristwatches and bracelets. Ground truth data sets have been manually computed by using an image graphic tool. Thus, ellipse positions and orientations have been acquired so that performance comparisons can be made between both filters and ground truth data. To evaluate and compare the filter performances the root mean square error (RMSE) is computed:

$$RMSE(\hat{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x(i) - \hat{x}(i))^2} \qquad (10)$$

Input images for both filters are the ones yielded by the skin segmentation algorithm described in Section II. State and measurement noise matrices were chosen to be the same either for the PF or the KF. In the case of the PF several experiments were carried out with different number of particles. An area based ellipse fitting method has been also applied to get the measures for the KF estimator. The prediction yielded by the KF is used to restrict the region where next measures are taken. A threshold operator combined with morphological filters is applied to the skin probability images to reduce the number of points where the ellipses are centered and thus, reduce the number of operations needed by the area based ellipse fitting process. The measures yielded by this method are very similar to the measures taken by the PF estimator.



(a)

(b)

Fig. 5. RMSE analysis for Particle and Kalman filters. (a) Orientation $\theta$ (b) XY Position

Both tracking techniques were implemented on a PIV 2.6 GHz with 512Mb RAM. The ideal number of particles is 200 for a computation time of 10 fps and an accurate response. Table I depicts RMSE for position and orientation for PF with different number of particles and for KF. It is shown that PF yields better

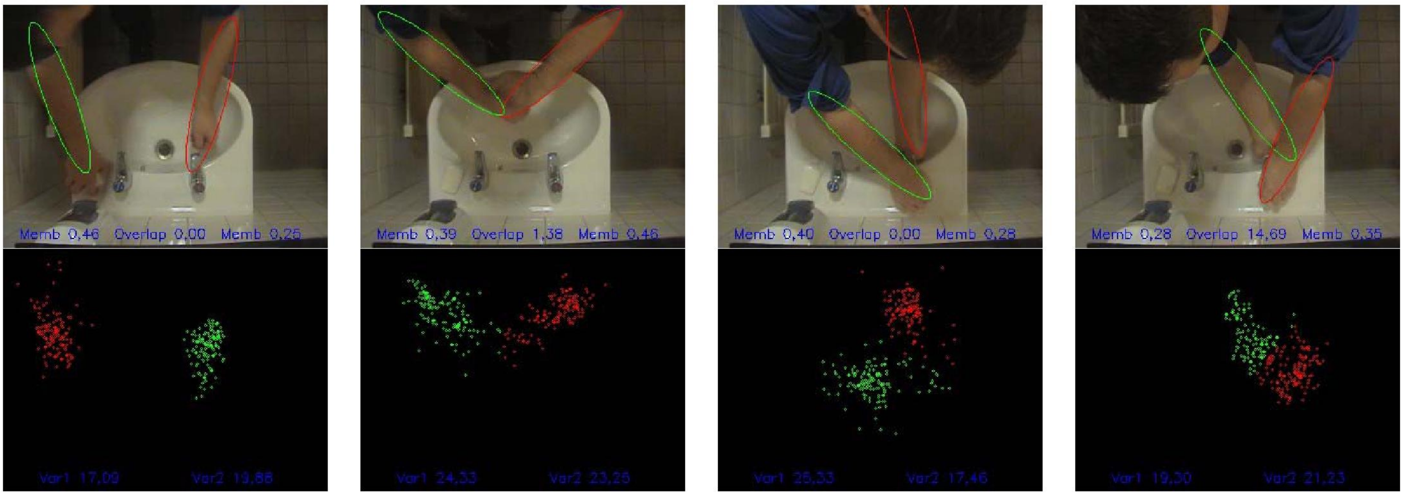

Memb 0,46 Overlap 0,00 Memb 0,25
Memb 0,39 Overlap 1,38 Memb 0,46
Memb 0,40 Overlap 0,00 Memb 0,28
Memb 0,28 Overlap 14,69 Memb 0,35
Var1 17,09 Var2 19,88
Var1 24,33 Var2 23,25
Var1 25,33 Var2 17,46
Var1 19,30 Var2 21,23

Fig. 6. TRACKING SEQUENCE. UPPER ROW: PARTICLE FILTER RESULTS. LOWER ROW: X-Y PARTICLES DISTRIBUTION

results for orientation as well as for position estimations. As long as the number of particles increases, the RMSE decreases and less time is needed to get stable estimations. On the other hand the computation time increases with the number of particles. The better performance of the PF can be explained by the multi-modal and non-linear nature of the problem to estimate. KF is able to predict the state of the system as long as it remains "inside the limits" of linearity. When both arms overlap or move too fast the KF fails in its predictions.

Figure 5 depicts the RMSE along time for the PF an KF. As can be seen PF yields better estimations for arms orientation and position. Also PF is robust to head occlusions and hands overlapping. It is able to maintain the estimation of the pose and the orientation for a few frames thanks to the supervision algorithm which adjust the filter response to the estimated situation of the system. It also performs better to quick movements of the arms, where the KF shows some inertia and sometimes loses the arms. The supervision algorithm reinitialize the estimation when occlusions or overlapping areas persist in time. In Figure 6 an example sequence where the results of the PF tracker are shown.

## V. CONCLUSIONS AND FUTURE WORK

A robust estimator of hands/arms position, orientation and velocity in a hand washing quality assessment system has been designed and tested. The proposed algorithm is based on a probabilistic multi-modal filter and is completed with a k-means clustering technique. Hands/arms are segmented by means of skin features and modeled by ellipses. The RMSE analysis has been used to describe and compare performances. The obtained results showed that the PF estimator performs better than the usual KF estimator. It is robust to partial occlusions and fast movements whereas KF is only able to track soft movements. The algorithm should be optimized in computing time, programming some of its critical parts in SSE2 or Altivec.

| # Particles | RMSE Orientation | RMSE Position |
|---|---|---|
| N=50 | 8.2956 | 12.9675 |
| N=75 | 8.2736 | 10.6263 |
| N=100 | 7.6524 | 9.8123 |
| N=150 | 7.8920 | 7.9388 |
| N=200 | 7.5046 | 8.5176 |
| N=250 | 7.3416 | 8.1302 |
| N=300 | 7.1035 | 7.5869 |
| Kalman | 25.3185 | 12.6010 |

TABLE I.

EFFECT OF THE NUMBER OF PARTICLES ON THE PERFORMANCE

## ACKNOWLEDGMENT

This work has been partially supported by Enterprise Ireland project CFTD/2006/216

## REFERENCES

[1] MRSA, "Methicillin-resistant staphylococcus aureus. guidance for nursing staff," in *Royal College of Nursing*, 2005.

[2] D. Fernández, F. Vilarino, J. Zhou, and G. Lacey, "A multi-class svm classifier ensemble for automatic hand washing quality assessment," in *In. Proc. of the BMVC*, 2007.

[3] J. Hoey, "Tracking using flocks of features, with application to assisted handwashing," in *In. Proc. of the British Machine Vision Conference (BMVC)*, 2006.

[4] M. Marrón, J.C. Garca, M.A. Sotelo, D. Fernández, and D. Pizarro, "Xpfcp: An extended particle filter for tracking multiple and dynamic objects in complex enviroments," in *In. Proc. of IEEE ICRA*, 2005.

[5] M. Storring, *Computer Vision and Human Skin Color*, PhD thesis, Faculty of Engineering and Sciencem Aalborg University, Niels Jernes Vej 14, DK-9220 Aalborg.

[6] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?," in *In Proc. of ECCV*, 1998.

[7] P. Chen and C. Grecos, "A fast skin region detector," in *In Proc. of IEEE VIE*, 2005.

[8] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28.

[9] M. Isard and J. MacCormick, "Bramble: A bayesian multiple-blob tracker," in *In. Proc. of IEEE ICCV*, 2001.

[10] M. Sanjeev, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.