People Tracking and Recognition using the Multi-Object Particle Filter Algorithm and Hierarchical PCA Method

David Schleicher, Luis M. Bergasa, *Member, IEEE,* Rafael Barea, Elena López

***Abstract* — This paper presents a method to detect, recognize and track people using mount cameras fixed on a building. The method consists of two independent stages. One is dedicated to detect and track any moving object within the image frame. The other one is in charge to discard any moving object that is not a human being. To perform the first task, a Particle Filter algorithm is used, in such way that it can perform the tracking of multiple objects. For the recognition stage a PCA *(Principal Components Analysis)* method is applied to several body parts (head, arms, etc) respecting their geometrical constraints. The performance of the system has been tested successfully. Some experimental results and conclusions are presented.**

***Keywords* — People detection, people tracking, components particle filter, multiple object tracking.**

I. Introduction

THE purpose of this paper is to present a method to detect, track and recognize people inside buildings, using computer vision, in order to monitoring human activities in an indoor environment. For that purpose, two main stages are defined; one dedicated to detect and track objects and the other dedicated to the recognition process.

For the first part, a Particle Filter (PF) algorithm is used. PF are widely used in tracking systems, as it is shown in [1]-[4]. General PFs are sequential Monte Carlo estimators based on particle representations of probability densities, which can be applied to any state-space model [4]. In order to track multiple objects, it is needed to solve the data association problem. It means to identify uniquely each tracked object with its own tracker. Several methods have been proposed for this purpose. The simplest one is the *nearest neighborhood* (NN) [5], which is based on using the closest observation to any given state to perform the measurement update step. Another method that is widely used is the *joint probability data association filter* (JPDAF). The goal of this method is to estimate the states by a sum over all association hypothesis weighted by he probabilities from the likelihood [6].

In our study, we chose to track each different object using a different PF. Due to the simple observation model used the computational cost is not significantly increased with the number of objects to be tracked.

For the recognition step, one of the most used methods is the *principal component analysis* (PCA) [7]. PCA tries to solve the recognition problem by reducing the dimensionality of the data (both training and sample data). Then, it keeps the most significant components that will be used on the decision phase. This method is widely used for face recognition. For cases where there is more than one class in the same training dataset, an improved method is the *linear discriminant analysis* (LDA) [8]. The objective of this method is to perform the dimensionality reduction while preserving as much as possible the separation of the different classes. When trying to identify the full body of human beans, several methods are applied. Some of them are based on applying recognition to several components of the human body instead of the complete image. In [9] a method for human body recognition is presented. They choose the head, legs, left arm and right arm as regions where apply *support vector machine* (SVM) classifiers. Then, the result of each individual classifier is taken as the input of another classifier, which output is the result of a "person" or "non person" decision.

Our approach is based on selecting several body regions with geometrical constraints. Once the object to identify is detected and located, PCA is applied to each individual region assuming that it is in fact a person. Each region will have both random dimensions and relative positions within the allowed margins. The output of each individual classifier is then introduced into a weighted function, which output will be the result of a "person" or "non person" decision as well. Then, it can be decided to track or not the detected object.

II. Detection and Tracking Stage

As mentioned before, this stage is based on a PF algorithm. The steps to follow on a generic PF are explained first. Then, the particular *observation* and

This work has been supported by grants DPI2002-02193 (SIRAPEM Project) from the Spanish Ministry of Science and Technology (MCyT) and grants GR/SAL/0860/2004 (TEAPEM Project) from the Community of Madrid Science and Technology Department.

David Schleicher is with Airbus Spain (Madrid). Spain (phone: +34-666244074, fax: +34-916241748; email: dsg68818@telefonica.net)

Luis M. Bergasa is with the Department of Electronics, University of Alcalá, Spain (phone: +34-918856569; fax: +34-918856591; e-mail: bergasa@depeca.uah.es; http://www.depeca.uah.es/personal/bergasa)

Rafael Barea is with the Department of Electronics, University of Alcalá, Spain (phone: +34-918856574; fax: +34-918856591; e-mail: barea@depeca.uah.es; http://www.depeca.uah.es/personal/barea)

Elena López is with the Department of Electronics, University of Alcalá, Spain (phone: +34-918856567; fax: +34-918856591; e-mail: elena@depeca.uah.es; http://www.depeca.uah.es/personal/elena)

prediction models used in the present work are described.

A. Particle Filter Algorithm

The PF is one of the so-called *Bayesian filters*. Its purpose is to estimate recursively the *posterior probability function:*

$$p(x_t \mid z_t)$$

, where x_t is the current state and z_t is the current observation. This function can be represented using a set of weighted particles:

$$\left\{(\alpha_t^0, \pi_t^0), \ldots (\alpha_t^N, \pi_t^N)\right\}$$

, where π_t^n represent the weight of each particle, which is obtained from the observation model, and α_t^n represent the particle state.

The algorithm follows the steps below:

1. A particle set is generated with an initial distribution.
2. Then, the observation step takes place and the weights are obtained (observation model):
$$\pi_t^n = p(z_t \mid x_t = \alpha_t^n)$$
Also, the cumulative probabilities are calculated:
$$c_t^n = c_t^{n-1} + \pi_t^n$$
Then, the weights are normalized by the maximum cumulative probability.
3. After that, a new set of particles is generated by resampling with replacement *N* times.
4. At the end of the recursive process the prediction step is applied. It is done by the evaluation of the prediction model:
$$p(x_{t+1} \mid x_t = \alpha_t^n)$$

The last step is the estimation of the current state:

$$x_t = \sum_{n=1}^{N} \pi_t^n \alpha_t^n$$

B. Observation Model

The observation model is based on the background subtraction technique. The objective is to identify any moving object on the scene taking as a basis that the camera is in a fixed position. The first step is to select the background of the scene (static scenario). Once we have stored the background scene, we take each frame of the sequence and obtain the absolute difference (pixel by pixel). The resultant image is then binarized. After an erosion and dilatation process, the resultant image is segmented to find all shapes that comply with certain geometrical constraints (size, aspect ratio, etc.). The result is showed on Fig 1.

Taking the geometrical characteristics of the observed regions, it can be defined the state vector to be used. It will be composed of 3 components:

$$x_t = \left(X_{centroid}, Y_{centroid}, Height\right)$$

The evaluation of the observation will be to calculate the Euclidean distance between the observed region and the particle to be evaluated. To do that, we need to know which region is associated to which particles.

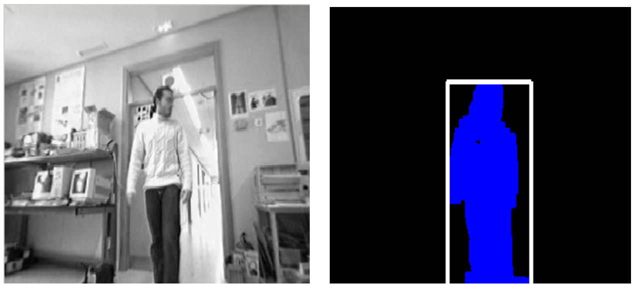

Fig. 1. *Left*: Captured scene. *Right*: A possible person detected by the process.

This is done by finding the *nearest* frame to each of the particles.

C. Prediction Model

Taking into account that the motion model that could be applied to people is strongly random, there is no appropriate deterministic model to be used. Then we chose to define a simple Gaussian and linear model to predict the next state:

$$x_t = x_{t-1} + a_{t-1} + w_{t-1}$$

w_t is the 0-mean Gaussian noise component. Sigma was empirically obtained. It was weighted by the estimated *Z* distance (from the camera to the object). a_t is the average state *speed*. It is defined as follows:

$$a_t = \frac{(x_t - x_{t-1}) + (a_{t-1}(\beta \cdot init - 1))}{\beta \cdot init}$$

Init is a variable that increases in a constant rate, up to a maximum value. It represents an importance factor, which represents the time elapsed since the object was detected. Then we give more weight to the previous cumulative average speed, depending on β factor.

III. Multi-object Management

As the method is designed for multiple objects, it has to be defined criteria to decide both the detection of new objects and the elimination of objects no longer detected.

To detect new objects on the scene, the candidate has to comply with the following requirements:

- *It shall not belong to any other already detected object Region of Interest (ROI).*
- *It shall comply with minimum and maximum size constraints.*

The selected criterion to eliminate unobserved objects is to monitor the following requirements as well:

- *The object centroid falls outside the image frame.*
- *The object is no longer observed and it was recently detected. (Assumed spurious detection).*

IV. Recognition Stage

Taking the regions associated to the objects as the basis, the other stage is the recognition one. For this purpose, a PCA algorithm, applied in a hierarchical way, is used. Due to the so many different orientations and poses of the human body, it makes difficult to apply any kind of *pattern* recognition method directly to the full image.

Instead of this solution, we chose to apply PCA method only to some specific regions within the detected object

ROI. These regions were located in the areas where, in a frontal view, the main human components are supposed to be found. We selected the *left* and *right arms* and the *head* as the most representative components.

Taking into account that these components can be placed in different positions respect to the main body and can have different sizes, it shall be defined a *variable* margin to look for them. These variable regions are shown in Fig 2.

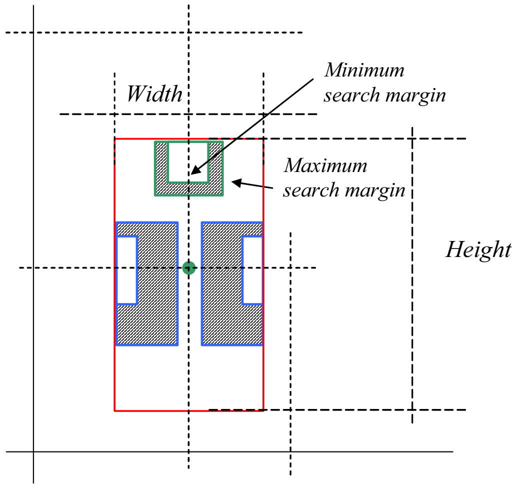

Fig. 2. Components search margins.

In order to apply PCA to these components, first a training data set has to be created. We used a training set composed of arms and heads indifferent posses and orientations.

The followed training process is explained below:

1. The first step is to resize all training samples $\Gamma_1, \Gamma_2, \ldots \Gamma_M$ to 24x24.
2. Then, the *covariance* matrix is calculated for each of the components (left arm, right arm and head) as follows:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n^T\Phi_n$$

where Φ_i are the training samples, normalized by the mean:

$$\Phi_i = \Gamma_i - \Psi \quad \text{, where:} \quad \Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

3. After that, the eigenvectors u_k and eigenvalues λ_k are calculated. Then, we choose the principal eigenvectors correspondent to the higher eigenvalues.
4. The following step is to represent the training vectors into the new base formed by the eigenvectors:

$$\Omega^T = [\omega_1, \omega_2, \ldots \omega_R] \quad \text{, where:} \quad \omega_k = u_k^T(\Gamma - \Psi)$$

Once PCA is trained, the recognition process to carry out, for each of the body components, is as follows:

1. Select the region to be analyzed.
2. Random samples at different sizes and positions are taken as we depicts in figure 3.
3. All the samples are resized to 24x24.
4. The correspondent vectors are represented on the new base in the same way as for the training process.
5. Then, the *Euclidean* distances between each sample vector and all the training set are obtained. For each sample the minimum distance is also acquired. Then, the global minimum is selected.

$$\{\in_{A,k} = \|A - \Omega_k\|\} \xrightarrow{\text{minimum}} \in_{A,\min} = \|A - \Omega_{\min}\|$$

$$\{\in_{B,k} = \|B - \Omega_k\|\} \xrightarrow{\text{minimum}} \in_{B,\min} = \|B - \Omega_{\min}\|$$

$$\{\in_{C,k} = \|C - \Omega_k\|\} \xrightarrow{\text{minimum}} \in_{C,\min} = \|C - \Omega_{\min}\|$$

$$\ldots\ldots\ldots\ldots \qquad \ldots\ldots\ldots\ldots$$

$$\{\in_{R,k} = \|R - \Omega_k\|\} \xrightarrow{\text{minimum}} \in_{R,\min} = \|R - \Omega_{\min}\|$$

$$\in_{\min} = \|Sample_{selected} - \Omega_{selected}\|$$

where A, B, C,… R are the vectors correspondent to the different samples.

6. Having the distances correspondent to the 3 human components, the last step is to decide if the object is a “person” or a “non person”. In order to do that we chose a weighted function that will give a result as a function of the 3 component classifiers:

$$\in_{total} = 2\in_{cara} + \in_{brazo\_izq} + \in_{brazo\_der}$$

If the result is above an empiric level, the object is recognized as “person”. Otherwise it is recognized as “non person”

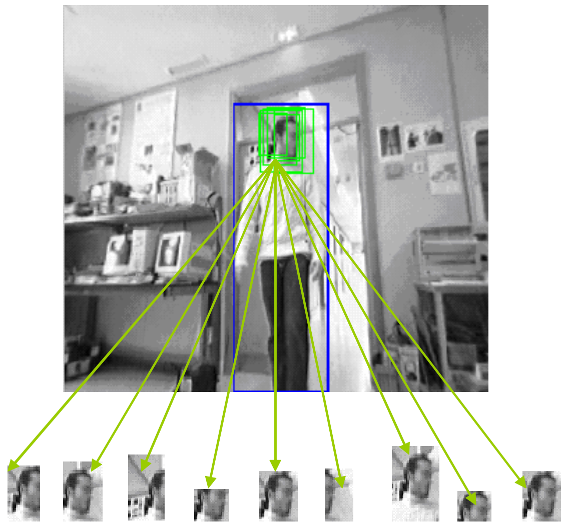

Fig. 3. Samples extraction process for the head component case.

V. Results

To test the performance of the system different video sequence were used. On each one there were up to 3 people present at the same time. There were occlusions, different poses, hidden objects, etc.

Respect to the tracking performance, it worked almost 100% of the cases when there were no object occlusions. The tracking was not lost also with low duration occlusions (see Fig 4). It was also taken into account on the algorithm cases where an object is hidden by parts of the background. In that case, the algorithm holds the last known position until the object appears again. It was also compared two alternatives when implementing the particle filter:

- One is to generate a new particle population each time a new object is detected.
- The other option is to keep constant the amount of particles independently of the number of objects.

It was observed that keeping constant the amount of particles, when increasing the number of objects, the performance was slightly degraded. However, taking into account that the whole algorithm is low time consuming, it is worthwhile to use the first option. On table 1 it is shown the success rate on tacking the detected objects (Successful tracking sample sequences vs Failed tracking sample sequences).

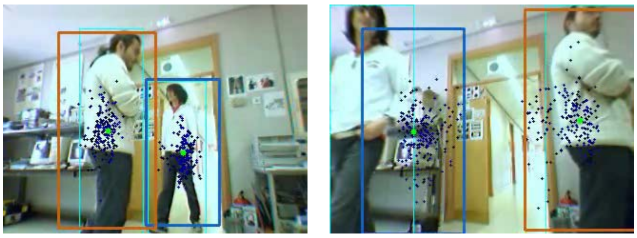

Fig. 4. Tracked people with occlusion.

Respect to the recognition process, it was observed that, having the object correctly located and tracked, people was positively recognized in almost all the cases. To indicate that a person has been positively recognized, a white circle is drawn around its centroid. The whole human analyzed were positively detected the most of the cases in frontal and back views (see Fig 5). Although the geometrical structure changes in an appreciable way for lateral views, the overall recognition process provides the correct result in the majority of the cases. The negative recognition behavior was tested also with positive results. On table 2 it is shown the success rate on the recognition process (Successful recognition frames vs Failed recognition frames).

VI. Future Work

In order to improve the performances of the tracking algorithm when having long term occlusions a stereo 3D system can be used. Using 3D system, the Z coordinate can be obtained more precisely, so that we can distinguish between far and near object in an accurate way.

VII. Conclusion

We have presented a method to detect, recognize and track people. A particle filter has been used to track multiple objects using a fixed camera. The observation method was based on background subtraction. To recognize the object detected as a "person" or "non person" the PCA method was applied in a hierarchical way (on some particular regions with geometrical constraints). Some preliminary experiments have been obtained with promising results showing that this method could be used in several applications such as surveillance, care, etc.

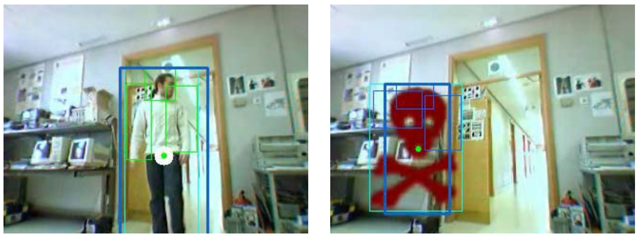

Fig. 5. (*Left*) Front positive recognition. (*Right*) Negative recognition.

Table 1: Tracking process success rates.

	Variable particle amount.	Constant particle amount.
No occlusion	98%	97%
Short-term occlusion	83%	77%
Long-term occlusion	32%	30%

Table 2: Recognition process success rates.

Front view	90%
Back view	78%
Lateral view	64%
Negative recognition	97%

References

[1] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," In Proc. European Conf. Computer Vision, 1996, pp.343-356, Cambridge, UK.

[2] S. Venegas, J. Knebel and J. Thiran, "Multi-Object Tracking using the Particle Filter Algorithm on the Top-View Plan," No ITS-02-04, 2004.

[3] Z. Khan, T. Balch and F. Dellaert, "Efficient Particle Filter-Based Tracking of Multiple Interacting Targets Using an MRF-based Motion Model," Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03), 2003.

[4] J.J. Pantrigo, A. Sánchez, K. Gianikellis, A. S. Montemayor, "2D Human Tracking by Efficient Model Fitting using a Path Relinking Particle Filter," Lecture Notes in Computer Science. Vol 3179/2004. pp 202 – 213.

[5] Y. Bar-Shalom and T. Fortmann. "Tracking and Data Association," volume 179 of Mathematics in Science and Engineering. Academic Press, 1988.

[6] R. Karlsson and F. Gustafsson, "Monte Carlo data association for multiple target tracking," invited paper to IEE Workshop on Target Tracking, Eindhoven, NL, 2001

[7] M. Turk, A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. Vol 3, No. 1. 71-86, 1991.

[8] A. Martinez and A. Kak, "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligenve, vol. 23, no. 2, pp. 228-233, 2001, (on-line).

[9] J. Reneau, "Tracking Using Intensity Gradients and Particle Filtering," Fall 2004 conference. Clemson University.

[10] G.Antonini, S.Venegas, J.P.Thiran, and M.Bierlaire, "Behavioral filtering of human trajectories for automatic multi-track initiation," Technical Report. Signal Processing Institute, EPFL, 2004.

[11] A. Mohan, C. Papageorgiou and T. Poggio, "Example-based Object Detection in Images by Components," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 4, April 2001.

[12] M. Pontil and A. Verri, "Support vector machines for 3D object recognition," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, no. 6, pp. 637-646, 1998.

[13] E. Stollnitz, T. DeRose, and D. Salesin, "Wavelets for Computer Graphics: A Primer (part 1)", IEEE Computer Graphics and Applications, vol. 15, no. 3, pp. 76-84, 1995.

[14] A. M. McIvor, "Background Subtraction Techniques," IVCNZ00, Hamilton, New Zealand, November 2000.