Particle filter tracking without dynamics

doi:10.1080/11762320701843499

Jaime Ortegon-Aguilar[a] and Eduardo Bayro-Corrochano[b]

[a] *Universidad de Quintana Roo, Engineering Department, Chetumal, Quintana Roo, Mexico*
[b] *CINVESTAV Unidad Guadalajara, Electric Engineering and Computer Science Department, Zapopan, Jalisco, Mexico*

Abstract: People tracking is an interesting topic in computer vision. It has applications in industrial areas such as surveillance or human-machine interaction. Particle Filters is a common algorithm for people tracking; challenging situations occur when the target's motion is poorly modelled or with unexpected motions. In this paper, an alternative to address people tracking is presented. The proposed algorithm is based in particle filters, but instead of using a dynamical model, it uses background subtraction to predict future locations of particles. The algorithm is able to track people in omnidirectional sequences with a low frame rate (one or two frames per second). Our approach can tackle unexpected discontinuities and changes in the direction of the motion. The main goal of the paper is to track people from laboratories, but it has applications in surveillance, mainly in controlled environments.

Key words: Omnidirectional vision, people tracking, particle filters, background subtraction.

INTRODUCTION

Omnidirectional Vision Systems are the topic of many research activities in recent years. The original idea for omnidirectional vision systems is to use cameras and mirrors to increase the field of view, as proposed by Rees (1970). Geyer and Daniilidis (2001) proposed a unified model for this kind of vision system.

People tracking is a major topic in computer vision. People tracking has applications in visual surveillance, human-computer interaction, etc. Major problems of people tracking are illumination changes, occlusions and unexpected motions of the subject. Particle Filters (PF) is a commonly used technique in people tracking. Traditionally, PF use system transitions to model the motion of the target. These transitions add flexibility in comparison to the Kalman filter. Despite there being lots of particle filter applications that succeed tracking targets, the modelling of dynamics represents a great challenge. Particularly, in people tracking with low frame-rate sequences, it is very difficult to model significant random jumps of subjects.

Particle filters had been widely used in very different areas like robotics, see a survey on this topic by Thrun (2002), and tracking. Arulampalam *et al.* (2002) made a survey on nonlinear/non-Gaussian tracking problems. They present several variants of the particle filter such as SIR, ASIR and RPF, comparing them to the standard EKF. Particle filters are particularly useful for visual tracking (Nummiaro *et al.* 2003, Pérez *et al.* 2002). Pérez *et al.* (2002) use a colour-based probabilistic tracking. They compare the colour content of candidate regions to a reference region, which is done with a colour likelihood based on colour histogram distances and the posterior with a Monte Carlo approximation. Nummiaro *et al.* (2003) proposed the use of particle filter with colour-based image features, specifically colour histograms. Their tracker takes advantage of the particle filter evaluating the image content only at the sample positions. They model the targets as ellipses and also compute the histograms assigning higher importance to those pixels that are close to the centre. Auxiliary Particle Filters (APF) were proposed by Pitt and Shephard (1999). APF generates particles from an importance density that depends on the most recent observations. It samples the posterior using the same importance density. Since APF relies on the most recent observations, it results on better priors and hence a better sampling of the posterior. Deutscher *et al.* (2000) introduce the concept of Annealed Particle Filter and this modified filter is an adaptation of simulated annealing to the particle filter. It uses different weighting functions at each step of the annealed process.

Background subtraction had been extensively studied in the computer vision community. Elgammal *et al.* (2002) use pixel intensity or colour to model the background. Their model keeps a sample of the intensity values for each pixel in the image and uses this sample to estimate the density function of the pixel intensity distribution. Haritaoglu *et al.* (2000) model the background variation with a bimodal distribution constructed from order

Corresponding author:
Jaime Ortegon-Aguilar
Email: jortegon@uqroo.mx

statistics of background values during a training period. Most algorithms for background subtraction are affected by illumination changes, and also must adapt to changes, like placing objects (should be part of the background).

Other works that use background subtraction as a step for tracking are those by McKenna *et al.* (2000) and Senior (2002). McKenna *et al.* (2000) track groups of people using background subtraction to get foreground regions. They classify these regions as *regions*, *people* or *groups*. They did not need to predict the motion of regions because they used sequences where the visual motion of regions were always small relative to their spatial extents. Senior (2002) presents a tracking algorithm which uses background subtraction and a so called high-level tracking. He first relates the tracks with the foreground regions and then applies a series of rules to solve ambiguities.

This paper presents a tracking algorithm for motion patterns that are hard to capture in dynamical models. The algorithm works with omnidirectional sequences, recorded in a laboratory, hence it does not compensate for illumination changes. Due to other requirements, sequences have low frame rate i.e., 1 or 2 fps. Consequently, targets can change size, orientation and location very abruptly and follow unpredicted trajectories. The targets are modelled as ellipses; using its covariance matrix, we take into account size and orientation changes. The likelihood is based on normalised colour histogram distances, because it is invariant to rotations and scales. The tracking is inspired on the particle filter, but the dynamical model is replaced with a background subtraction step and association of regions with tracks. Our algorithm is able to track people in omnidirectional sequences with low frame rates, where subjects move significantly and randomly between consecutive frames.

The paper is organised as follows: the second section gives a brief introduction to particle filters, the third section presents the colour model, the fourth section shows the background subtraction, and the fifth section depicts the modified version of particle filter. In the sixth section, experiments are shown. Finally the seventh section is devoted to conclusions.

PARTICLE FILTERING

The particle filter evolved from the work by Isard and Blake, CONDENSATION (1998). It was developed to track objects in clutter. The particle filter requires the definition of two elements: a data likelihood term and a dynamical model. The first one evaluates the likelihood of the current observation given in the current object state. The dynamical model takes information about the prior on the state sequence and helps to predict the new state.

Assume that the state of the tracked object at time t is denoted by x_t, its history is $X_t = \{x_1, \ldots, x_t\}$. The vector $Z_t = \{z_1, \ldots, z_t\}$ denotes all the observations z_i up to time t. In the particular case of tracking, z_t represents a set of image features at time t. The goal is to approximate the posterior $p(x_t|Z_t)$ of the probability distribution. The key idea in the particle filtering is to approximate the probability distribution (and consequently the posterior) by a weighted finite set of samples, the particles. Denote a weighted set of N different samples as $S = \{(s_t^{(n)}, \pi_t^{(n)})|n = 1, \ldots, N\}$. Each of these samples $s^{(i)}$ represents a possible object state, with an associated weight $\pi^{(i)}$. The likelihood of being the true location of the target for a particular sample (state) is represented by its weight. The weights are normalised such that $\sum_{i=1}^{N} \pi^{(n)} = 1$.

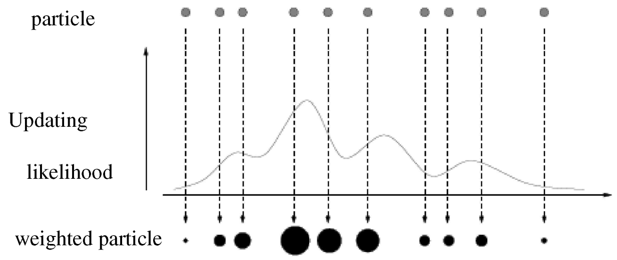


Figure 1 Illustration of the Particle Filter. The black points represent the weighted particle, the search is misled by local maximum.

The posterior $p(x_t|Z_t)$ can be expressed recursively applying Bayes law as,

$$p(x_t|Z_t) = k_t p(z_t|x_t)p(x_t|Z_{t-1}). \tag{1}$$

The distribution $p(x_t|Z_{t-1})$ is obtained from the posterior $p(x_{t-1}|Z_{t-1})$ at previous time $t-1$ by marginalising over x_{t-1},

$$\begin{aligned} p(x_t|Z_{t-1}) &= \int_{x_{t-1}} p(x_t, x_{t-1}|Z_{t-1}) \\ &= \int_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1}), \end{aligned} \tag{2}$$

where the chain rule was applied and $p(x_t|x_{t-1})$ is the dynamical model.

The evolution of the sample set is obtained as follows. All the particles are moved independently according to a system transition (dynamical) model. Then all the samples are weighted using a measurement density $p(z_t|x_t)$, such that $\pi_t^{(n)} = p(z_t|x_t = s^{(n)})$. The mean state is estimated at each time by $\mathrm{E}[S] = \sum_{n=1}^{N} \pi^{(n)} s^{(n)}$.

The PF reduce the computational cost by searching only those regions where the object is predicted to be by the model and samples. They are robust trackers because they model the uncertainty. See Figure 1.

The annealed particle filter approach uses a series of weighting functions $w_0(X)$ to $w_M(X)$, each w_m differs only slightly from w_{m-1}. The function w_m must be broader than w_{m-1}, because it must search a larger region. The i-th annealing run is performed with the predictions made by the previous run and using the function w_i to assign weights to the particles. See Figure 2.

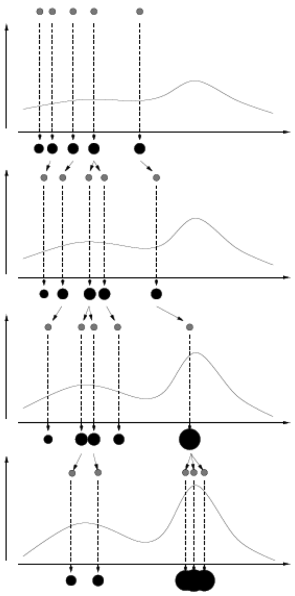


Figure 2 Illustration of the Annealed Particle Filter. The black points represent the weighted particle, through the layered search the particle set gets closer to the global maximum.

COLOUR MODEL

The tracking algorithm was designed to work in a colour context. There are a lot of different colour space models like the RGB (red, green, blue), the CMYK (Cyan, Magenta, Yellow, blacK), HSV (Hue, Saturation, Value), etc. The HSV model is used in the algorithm because it is less sensitive to light changes. However, the colour is only reliable when the saturation and value (brightness) are not too small.

We use colour distributions to model the information since they provide robustness against rotations, scales and partial occlusions. The distributions are discretised (N bins) and normalised. The target regions have histograms (discrete colour distributions) with N bins. In the following, N_h, N_s and N_v will represent the number of bins used for hue, saturation and value respectively. The histograms used are squared for the hue and saturation larger than two thresholds set to 0.1 and 0.2 respectively, with $N_h N_s$ bins. This histogram provides information about the colour of the target, but allows black and white being considered the same. Hence, the value information from the remaining pixels is also necessary, but with less sensitivity. It results in the use of histograms of $N = N_h N_s + N_v$ bins. For our experiments we use $N_h = N_s = N_v = 10$, the resulting histograms have $N = 10 \times 10 + 10 = 110$ bins. An example of the resulting histogram is presented in Figure 3.

There are some common distance measures for histograms like the intersection of histograms. As with Pérez *et al.* (2002) and Nummiaro *et al.* (2003), we use the Bhattacharyya similarity coefficient to compute the distance between histograms. This distance is used to favour colour histograms that are closer to the reference histogram. It is defined as follows:

$$\rho(p,q) = \sum_{i=1}^{N} \sqrt{p(i)q(i)} \tag{3}$$

$$d(p,q) = \sqrt{1-\rho(p,q)}, \tag{4}$$

where p, q are two histograms, $p(i)$ and $q(i)$ represent the i-th bin of the respective histogram and $d(p,q)$ is the distance between the histograms p and q.

BACKGROUND SUBTRACTION

Next, the background subtraction algorithm will be explained. The background model is computed with an algorithm similar to that by Haritaoglu *et al.* (2000). Haritaoglu *et al.* (2000) modelled the variation of the background with a bimodal distribution build with order statistics of the pixel values over a training period. The background model represents each pixel with three values: minimum, maximum and the greater difference between consecutive frames over the training period. The model updated following certain rules. When a certain amount of time elapses, the model is updated completely. Each pixel is classified as background or foreground using the previous values. Given the minimum (M), maximum (N) and the greater differences between frames (D), the pixels x of

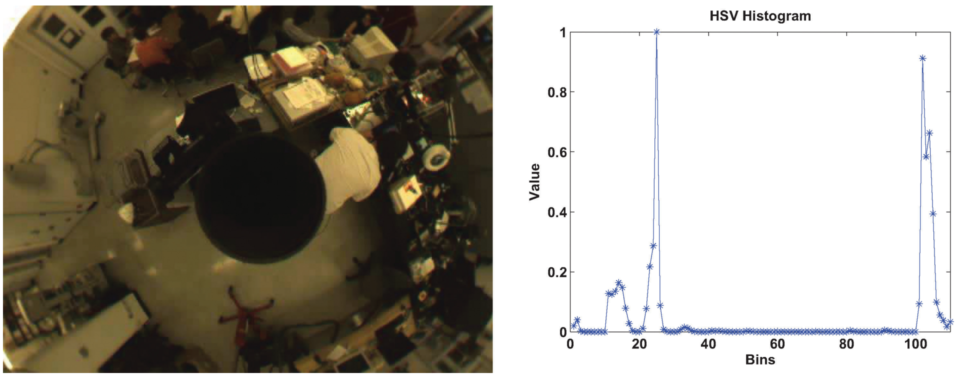


Figure 3 One image and its respective HSV histogram. The histogram has 100 bins for hue x saturation and 10 bins for value.

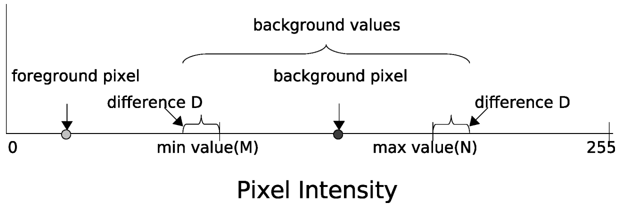


Figure 4 Graphic of the pixel selection as background or foreground. Pixels between (M-D) and (N+D) are considered as background.

an image I will be foreground if:

$$|M(x) - I(x)| > D(x) \ or \ |N(x) - I(x)| > D(x) \quad (5)$$

However, in order to increase the speed and reduce the memory space our algorithm does not keep a difference for each pixel, but uses a global one. Another difference is that our model is updated dynamically and independently for each pixel, which is updated depending on the number of times it keeps in range from the current value. Figure 4 graphically presents (5).

Figure 5 shows the background model. It is worth mentioning that the background model is computed in the grey scale colour model. Nevertheless, the tracking uses full colour information in the HSV colour model. The training period considered is 27, but it can be 81 images. For every new frame, after getting foreground regions, a morphological closing is made and small regions are discarded considering them as false positives.

TRACKING ALGORITHM

Before completely getting into the algorithm details, it is necessary to explain the modification to the particle filter. The used sequences have a low frame rate, and as a consequence the targets can move significantly between frames. Also, the targets can have unexpected motions. This results in a poor dynamical model for the particle filter. We propose the use of background subtraction techniques to get the predicted location of tracked regions for the filter. That is, the foreground regions not only represent regions that are moving but possibly targets that are being tracked as well. We use a rule that associates the foreground regions with the regions being tracked. This *association rule* says:

> *'If the distance between the centre of the tracked region and the perimeter of the foreground regions is less than a given threshold then the regions are associated and vice versa.'*

This association rule is described as

$$A \circ B \leftarrow \begin{cases} true \ \ if \, dist(A,B) \leq Th \\ false \ \ if \, dist(A,B) > Th \end{cases}, \quad (6)$$

where $\circ$ stands for association, $dist(A, B)$ stands for the distance in pixels from the centre of region A to the nearest point of region B and Th is a threshold for such association. The threshold for the experiments is set to the height or width of the target region. The safest case is to set the threshold so the search region is the whole image. In the latter, the annealed approach will increase not only the processing time but also the possibility to track the region correctly.

This way, the locations where the tracked region is more likely to be can be obtained without the need of dynamical models. This rule is graphically presented in Figure 6.

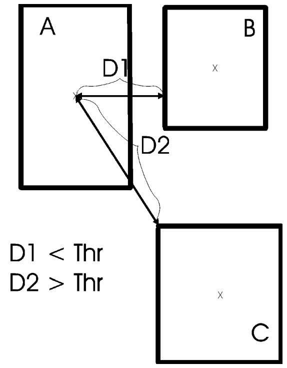


Figure 6 The region A and B are associated because their distance is lower than the threshold, but A and C are not associated because their distance is greater than the threshold.

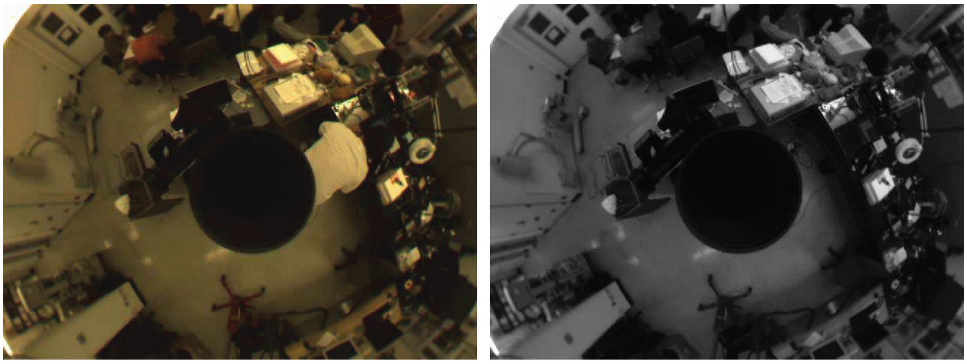


Figure 5 Background model. a) Original Image. b) Computed background.

The regions are modelled as ellipses using its covariance. By using the covariance matrix, we are able to change the orientation and scale of the ellipses. Before the samples (ellipses) are weighted, we rotate and scale the covariance matrix. Then we compute the correspondent histograms and weights for the transformed regions. One property of the covariance matrix is that it can be multiplied by a rotation matrix to get the effect as if the data were rotated and also it can be scaled in the same way. This property is written as follows:

$$\mathrm{cov}(RX, RX) = R\mathrm{cov}(X, X)R^T. \tag{7}$$

That is, computing the covariance matrix of points X rotated with a matrix R as $\mathrm{cov}(RX, RX)$ is the same as rotating the covariance matrix, $R\mathrm{cov}(X, X)R^T$. This also applies to scaled points, $\mathrm{cov}(SX, SX) = S\mathrm{cov}(X, X)S^T$. Using this property, the algorithm modifies the regions (samples) not only in their location but also in orientation and scale. The algorithm needs to select the pixels that are part of the region. Recalling that regions are ellipses, the covariance matrix is used to discern which pixels belong to the region. A region R with covariance C is then computed as a set of pixels $\mathbf{x}$,

$$R = \{pixel\,\mathbf{x} \mid C\mathbf{x}C^{\mathrm{T}} < \gamma\}, \tag{8}$$

where γ is a threshold to control the size of the region. It is set to a value of 0.85, with it the size of the ellipse and bounding box of the foreground region are almost the same.

To assign weights to each sample region, the Bhattacharyya coefficient is computed between the histogram of each sample and the histogram of the reference region using equation (4); this last histogram will be called a target histogram. As it was noted, the algorithm favours colour histograms that are close to the target histogram, hence it gives more weight to those histograms with small Bhattacharyya distance. The weights are assigned using a Gaussian function as follows:

$$\pi^{(n)} = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{d(p^{(n)},q)_n^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(1-\rho[p^{(n)},q]}{2\sigma^2}} \tag{9}$$

where $d(p^{(n)}, q)_n^2 = [1 - \rho(p^{(n)}, q)]$ is the distance corresponding to the n-th particle $p^{(n)}$ and the target histogram q.

The tracking is initialised with the background subtraction, taking the foreground regions as the targets to track. The background model is computed with the algorithm depicted in the fourth section. Ellipses are used to model these regions with their covariance matrix. Note that using these matrices, the algorithm does not force the ellipses to be vertical or horizontal as most algorithms do.

At each iteration of the algorithm, a new frame from the sequence is used; then the foreground regions are extracted using a background model previously computed. The foreground regions are associated with the current tracks using the association rule previously stated. All the foreground regions associated to the same track are fused and their centroid is considered as the predicted location for the track. In case no regions are associated, the samples will be drawn frm the whole image. From here, given the locations, the algorithm is able to draw samples about them, which are ellipses, maybe with different rotation and scale than the original one. Together, all previous steps are similar to propagate the sample set with the dynamical model.

Starting at this point, the algorithm follows the same steps as if it is an annealed particle filter. We already have an initial sample set. Compute the histogram of the target region. Next, compute the Bhattacharyya coefficient for each (modified) sample using equation (4). Assign weights to each sample with equation (9). Select only the n particles with greater weights, and for the $k-th$ annealed run, draw particles around the centroid of the previously selected particles, again weight each particle according to equation (9) and select those particles with greater weights. After the end of the last annealed run, compute the predicted region as the expected value of the particles:

$$\mathrm{E}[S] = \sum_{n=1}^{N} \pi^{(n)} s^{(n)}. \tag{10}$$

The complete algorithm is presented in Figure 7.

EXPERIMENTS AND RESULTS

The authors present the experiments with an omnidirectional sequence. Also, the results of the proposed algorithm are compared with a standard particle filter.

```
1. Compute the foreground regions of the frame
2. Associate the regions with the tracks using equation (6)
3. Fuse all associated regions to the same track
+ For the first annealed run,
   4. Draw $N$ particles around the centroid of the new region
   5. Weight each particle according to equation (9)
   6. Select only the $n$ particles with greater weights
+ For the k-th annealed run,
   4. Draw $N$ particles around the centroid of the selected particles
   5. Weight each particle according to equation (9)
   6. Select only the $n$ particles with greater weights
+ After the end of the last annealed run
   7. The predicted region will be the expected value of the particles
                    $\mathrm{E[S]} = \Sigma_i^n \pi(i)s(i)$
```

Figure 7 The tracking algorithm. It uses an annealed particle filter approach and background subtraction step.

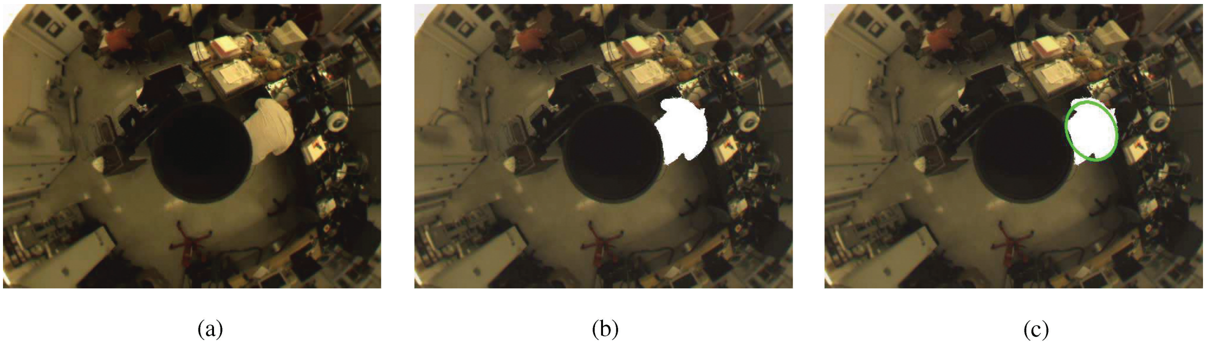


Figure 8 Target region (ellipse). a) Original Image b) Foreground region. c) Overlapped region and ellipse.

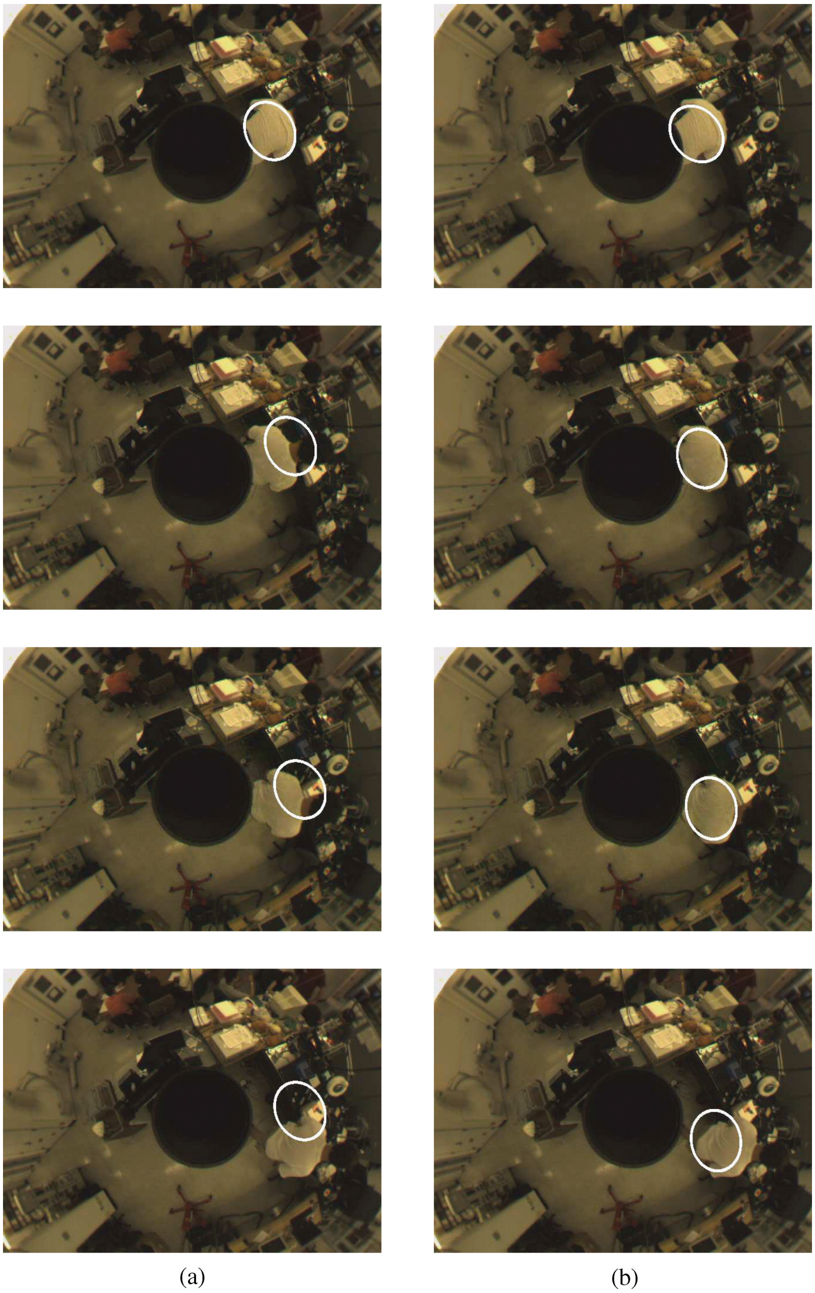


Figure 9 Comparison. a) Classical particle filter. b) Proposed algorithm. Note that the classical algorithm quickly loses the region.

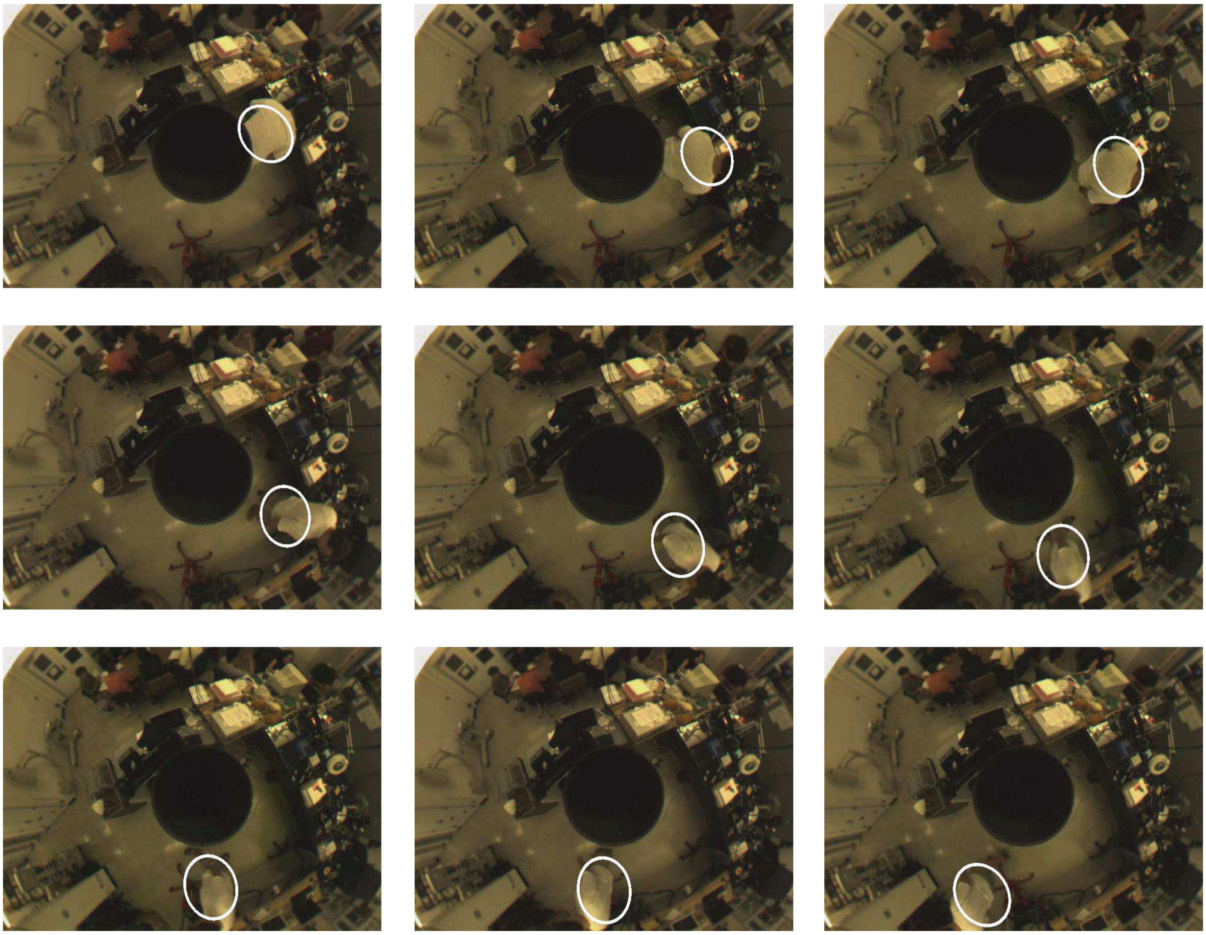


Figure 10 Complete Sequence. Frame 1, 7, 10, 14, 17, 20, 23, 26 and 29.

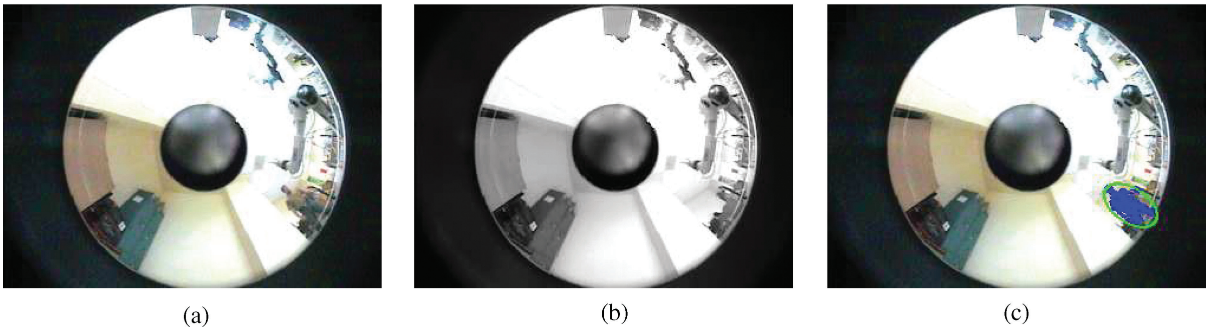


Figure 11 Second sequence of images. a) Original (first) image. b) Computed background. c) Foreground region and corresponding ellipse.

Omnidirectional sequences are selected because they do not fulfil the spatial assumptions regarding vertical and horizontal projections made in some papers (McKenna *et al.* 2000). Also they provide a larger field of view than projective cameras, but loses some resolution. Note that the frames suffer high order distortions due to the mirror reflection. For additional information about omnidirectional vision systems the readers can see the work by Geyer and Daniilidis (2001).

The sequence used have a low frame rate, only 2 frames per second. The sequence was grabbed in a laboratory with the camera placed in the centre of the room. As the reader can see, the target rotate almost 120 degrees about the centre of the camera. The region to track is presented in Figure 8. Figure 8 presents the foreground region used to compute the target ellipse and corresponding histogram.

The comparison with a standard particle filter is presented in Figure 9. In this figure are shown the frames 1, 5,

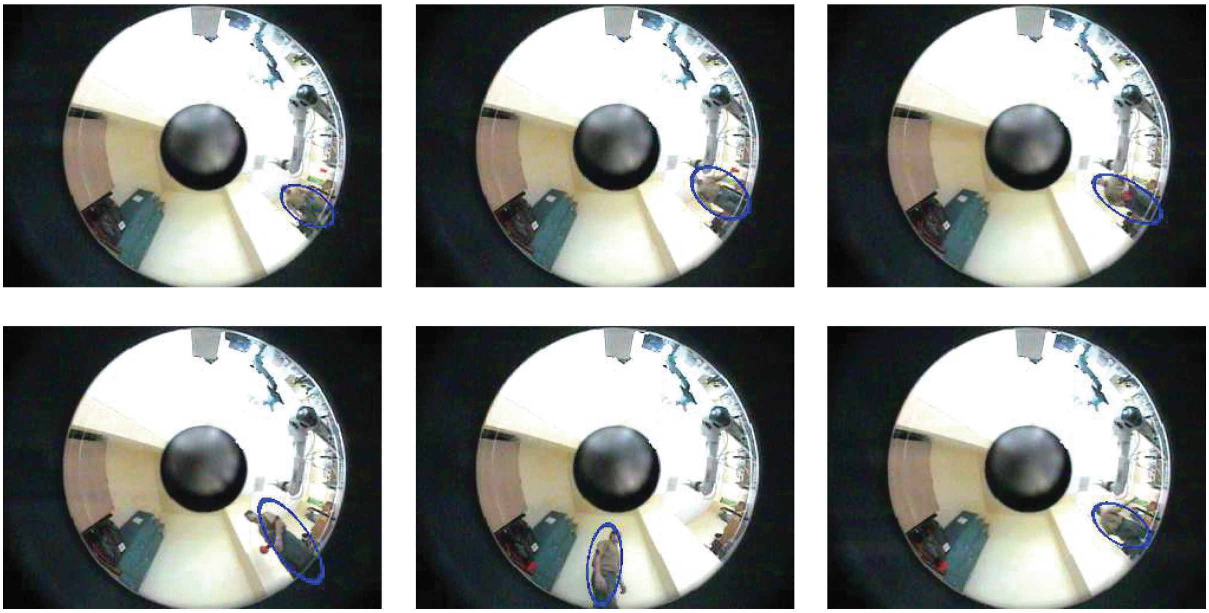


Figure 12 Complete sequence. Frame 1, 3, 4, 6, 11 and 14.

7 and 12, as the reader can see the standard algorithm fails by frame 5, where the move of the target is significantly greater than the movement in previous frames. Note that dynamical models can hardly follow erratic trajectories of targets. Moreover the region does not rotate or scale hence it is more difficult to find the correct region. As a result the weight of the proposed region decreases rapidly. Our approach is certainly more robust, because it relays on the likelihood of the current observations given the current object state. It does not take into account any dynamical model whatsoever, using instead foreground regions to update the locations of the samples (regions).

Figure 10 shows the evolution of the tracking sequence, from starting frame to frame 30 where the target left the field view of the camera. The frames shown are 1, 7, 10, 14, 17, 20, 23, 26 and 29. It is easy to see that the movement at some frames is large and that the target rotates and changes its scale significantly. Even when this happens, the proposed algorithm is able to track the target regions all along the sequence.

A second sequence is presented using other omnidirectional system, but it also has low frame rate. The system has a parabolic mirror. Figure 11 shows the first image of the sequence; the computed background, the selected foreground region and corresponding ellipse. Finally, Figure 12 presents the evolution of the sequence.

CONCLUSIONS

This paper presents an alternative to tracking challenging omnidirectional sequences. This alternative is an algorithm able to track targets even when they rotate, change scale and experiment significant changes in location and/or occlusions. The algorithm models the target regions as ellipses; these ellipses are described using the covariance matrix of the target, as a result the ellipses can be rotated or scaled. The proposed algorithm is especially useful in situations where the frame rate is very low due to physical restrictions on the system, bandwidth or another processing. The proposed algorithm uses a modified version of the particle filter where the dynamical model is replaced by a background subtraction and data association.

ACKNOWLEDGEMENTS

The authors would like to thank to Cinvestav, especially to Cinvestav Guadalajara and Conacyt for the funding and resources for the work.

REFERENCES

Arulampalam MS, Maskell S, Gordon N, Clapp T. 2002. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–88.

Baker S, Nayar S. (November 1999) A theory of single-viewpoint catadioptric image formation. *Intl J Comp Vis*, 35(2):1–22.

Deutscher J, Blake A, Reid I. 2000. Articulated body motion capture by annealed particle filtering. In: Proceedingd Conference Computer Vision and Pattern Recognition, vol. 2, p. 1144–49.

Elgammal A, Duraiswami R, Harwood D, Davis LS. 2002. Background and foreground modeling using nonparametric kernel density for visual surveillance. In: Proceedings of the IEEE, vol. 90, p. 1151–63.

Geyer C, Daniilidis K. 2001. Catadioptric projective geometry. *Int J Comp Vis*, 45:223–43.

Haritaoglu I, Harwood D, David LS. 2000. W4: Real-time surveillance of people and their activities. *IEEE Trans Pattern Anal Mach Intell*, 22(8):809–30.

Isard M, Blake A. 1998. Condensation–conditional density propagation for visual tracking. *Int J of Comp Vis*, 1(29):5–28.

McKenna SJ, Jabri S, Duric Z, Rosenfeld A, Wechsler H. 2000. Tracking groups of people. *Computer Vision and Image Understanding: CVIU*, 80(1):42–56.

Nummiaro K, Koller-Meier E, Gool LJV, et al. 2003. An adaptive color-based particle filter. *Image Vision Comput*, 21(1):99–110.

Pérez P, Hue C, Vermaak J, Gangnet M. 2002. Color-based probabilistic tracking. In: *ECCV02*, vol. I, p. 661–75.

Pitt MK, Shephard N. 1999. Filtering via simulation: Auxiliary particle filters. *J Am Stat Assoc*, 94(446):590–99.

Rees DW. 1970. Panoramic television viewing system, in US Patent Num. 35054665.

Senior AW. 2002. Tracking with probabilistic appearance models. In: Proc. ECCV Workshop on Performance Evaluation of Tracking and Surveillance Systems (PETS), p. 48–55.

Thrun S. 2002. Particle filters in robotics. In Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI).

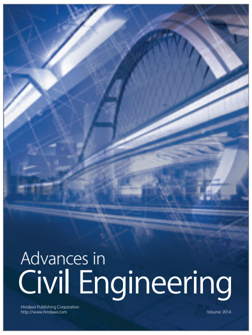
Journal of
Engineering

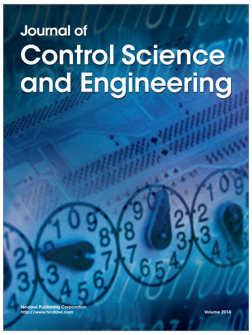
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

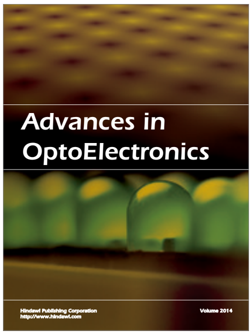
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

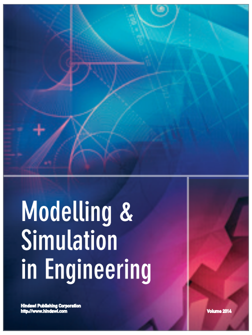
Journal of
Robotics

Journal of
Electrical and Computer Engineering

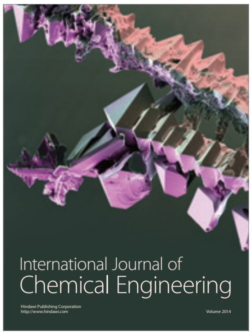

Advances in
OptoElectronics

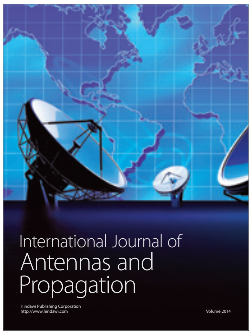
VLSI Design

International Journal of
Navigation and Observation

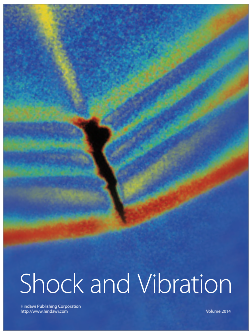

Modelling & Simulation in Engineering

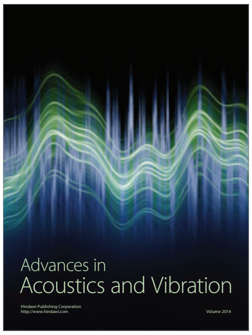
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration