People detection and tracking using the Explorative Particle Filtering

Jamal Saboune and Robert Laganiere
School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5
jsaboune,laganier@site.uottawa.ca

Abstract

Automatic people detection and tracking is a very essential task of video surveillance systems. It can improve a system's performance in important fields such as security, safety, human activity monitoring etc. In this paper we present a novel approach for people detection and 3D tracking. Our method is based on a human upper body 3D model and a likelihood function to evaluate its presence in a certain region of the scene. We then find the maxima of this function using a modified particle filtering algorithm which we call Explorative Particle Filtering (ExPF). We designed this algorithm in a way to guarantee a multiple objects tracking and a good estimation of their positions when using a small number of particles. Our technique is generic and simple as no dynamic models nor trained features models (color, shape etc.) were used. We also show some tracking results from video surveillance feeds in order to illustrate our approach.

1. Introduction

Automatic detection and tracking of people in real world surveillance videos is a very important task to perform and can be used to improve automated surveillance systems in many aspects. In fact it can help securing a specific area by estimating a crowd's density, counting the number of people entering or leaving this area or detecting unusual behaviours for example. Detecting people presence in front of a showcase, a poster or shop can also be useful to evaluate the efficiency of a marketing strategy. An automatic people tracking system can also contribute to the monitoring of human activity and people interaction at home and thus can detect intrusion or abnormal activities.

In order to be efficient, the detection and tracking module must be generic as the movement detection should not be restricted to specific situations or especially arranged environments. The system conceived is also supposed to be little sensitive to illumination conditions, clothing and background changes. It is meant to detect people entering the scene from any side without any previous knowledge of the shape or appearance of the detected person. Our idea is to develop an algorithm able to detect and track the people in a scene without the need to use expensive or sophisticated hardware.

Individual detection modules, previously developed for surveillance systems, use trained models of the human body appearances or shapes and try to distinguish image regions containing people using classification algorithms. The Pfinder system [25] uses blobs contours information, statistical modelling of colors and shapes to track the movement of a single person in the scene. Using frequency changes in the image Marana *et al.* [15] estimate textures and apply a trained Kohonen maps network in order to estimate a crowd density. However, high frequency variations in the background affect textures estimation and as a result reduce the estimation accuracy. McKenna *et al.* [16] model the color distribution of the moving objects using adaptive Gaussian mixture in order to track people. Nakajima *et al.* [18] tracker uses Support Vector Machines to classify foreground objects represented by shape and color features. A hierarchical template matching based on blobs contour and distance transforms is applied in [4] to detect pedestrians. As we are looking to detect and track people in real world situation all these methods would act poorly, due to the frequent changes in color features resulting from changing illumination and different camera angles. On the other hand, as the human movement is non rigid, the entire body shape can change frequently, thus the shapes training set cannot be reduced to a small set of configurations. Also, the shape and color models would fail to detect people in occlusion situations (people carrying boxes or bags, body parts occluded by other people etc.).

In order to overcome these problems many tracking systems have been developed using multi-camera information. Tsutui *et al.* [21] use multiple camera and optical flow estimation in order to calculate the velocity and the 3D position of a moving object. However this method is able to track a single target which restricts its application. Stereo vision techniques combined with shape matching [5, 12], color information matching [10, 17] or probabilistic models [17, 6] can determine whether objects observed in different views are the same. After detecting the different objects in the scene, their counting and tracking become easy to accomplish. Meanwhile, matching objects in many views is of high computational complexity and requires an extensive calibration work.

Instead of using global features of the human body some methods try to extract some features from the video and classify them as local features of a human body. Thus, by detecting some body parts they detect a person's presence. Haar-like features developed for face detection [23] have been used with an AdaBoost classifier [24] or in an extended way [11] to detect people. Histograms of oriented gradients (HOG) [3], local edge parts or edgelets [26, 19, 27] are other local features introduced for people detection. Lu *et al.* [13] use HOG, color and shape models and a particle filer based tracker to detect multiple hockey players. These discriminative techniques detect body parts with a high efficiency. On the other hand, they require large training data sets and complex image processing tasks.

Detection and tracking of multiple humans can also be considered as a Bayesian inference problem. In fact, people detecting in the image can be viewed as a problem of finding the state vector, describing the number and positions of people in the scene, which fits the best to the observation provided by some image features. Zhao *et al.* [29] use 3D human shape model, appearance model and a Markov Chain Monte Carlo (MCMC) technique to detect and track multiple humans in crowded scenes. Despite being efficient this approach is of a large complexity and requires the extensive knowledge of the scene entrances, exits and its geometry. One of the most efficient Bayesian estimators is the Condensation algorithm [7] also known as particle filter. This algorithm is designed to handle multimodal and non-Gaussian probability densities and can model uncertainty. Thus, we opted to develop a solution based on this algorithm. In section 2 we describe the Condensation algorithm and its application to tracking. We also discuss the necessity to adapt it to multiple objects tracking. The state vector and the function we use to evaluate its likelihood are exposed in section 3. Section 4 outlines the modified particle filter method we conceived. Results and discussion are shown in section 5.

2. Particle filtering and object tracking

The Condensation or particle filter algorithm was conceived to track an object in video images. In a particle filter context, a given configuration of the state vector X, we are willing to estimate, is called *particle*. The likelihood $P(Z_k/X^m)$ of an observation (or the history of observations) Z_k given a particle X^m is called *weight*. The goal of any Bayesian state estimator is to find the best fitting configuration of the state vector at time t (X_t) given all the observations until time t $(Z_{0:t})$. This can be done by finding the maximum of the *a posteriori* density $P(X_t/Z_t)$. However, it is not always possible to calculate this density in a direct way. The idea of the Condensation technique is to represent this *a posteriori* density using a set of sampled weighted particles. This set of particles is established using the likelihood function and the a priori density $P(X_t/Z_{t-1})$ calculated through the process dynamics and the set of samples used at time $t-1$ to represent $P(X_{t-1}/Z_{t-1})$.

The original particle filter algorithm was designed for a single object tracking. To reach that goal, it tends to concentrate and create the particles used for estimation, around the *a posteriori* density maxima estimated at the previous time step. Thus, when used with a small number of particles, most of the created particles would be close to the previously estimated state vector. As a result, this algorithm is unable to detect a new object entering the scene and would badly recover after some erroneous observation (occlusion for ex.). In order to fairly represent the *a posteriori* density in these situations a larger minimum number of particles are needed. But, when dealing with large dimension state vectors this number becomes big and makes the algorithm practically inapplicable. Many algorithms have been proposed to reduce the complexity of the Condensation algorithm by reducing the minimum number of particles needed to have a good estimation [28, 22]. These approaches are designed for a single object tracking, and like the original particle filter, tend to exploit the zone of the state vectors search space, labelled as most probable, rather than to explore the entire search space. Thus, the ability of the tracker to detect newly appearing objects is still reduced. Saboune *et al.* [20] introduced a modified particle filtering for 3D human motion capture. The proposed Interval Particle Filtering algorithm reduces the number of particles needed and overcomes the particles degeneration problem by introducing constant particles. This approach is interesting but needs to be adapted to multi object tracking.

Other methods based on particle filtering were introduced for multi object tracking and were capable of dealing with objects newly appearing. Previously presented techniques [14, 8] use joint likelihood functions and a joint state vector composed of the different state vectors describing the different objects. In fact, the size of the joint state vector depends on the number of objects detected; when this number grows, the joint state vector size enlarges and thus a greater number of particles are necessary to have a good estimation, which makes this method computationally complex. Koller-Maier *et al.* [9] use an individual classic Condensation tracker for each object and then combine the densities resulting from each of these trackers to get a global density describing the entire process. Their method is based on the assumption that all objects are detected with the same accuracy. If this property is not verified, which is the case when tracking people at different depths with different shapes and sizes, the sample set they use may degenerate and the tracking will fail. Actually in this case, all the particles would be concentrated around the particles having the larger weights.

Our idea is to use a simple likelihood function to evaluate the presence of a single person in the image. We then represent the *a posteriori* density using a reduced number of particles, by applying a modified particle filtering algorithm adapted to multiple objects detecting and tracking. Our algorithm which we call Explorative Particle Filtering (ExPF), will be exposed in section 4.

3. The likelihood function

We aim to develop a simple likelihood function which evaluates the presence of a person in the images. We then estimate the presence of people in the scene by detecting the maxima of this function using a particle based approach. As we are dealing with feeds provided by surveillance cameras, the method we apply should be able to handle low resolution and low frequency feeds. Thus, we avoid using any feature which is sensitive to noise or which requires a high resolution image to be tracked. The evaluation function should also be robust to partial occlusions and illumination changes. In order to satisfy all these constraints we opted to conceive a function based on the foreground silhouette images. These images are extracted by subtracting the background, applying a threshold then a median filter to reduce the noise, and a HSV shadow elimination filter [2].

Our approach is also based on using a 3D model, simulating the upper body human shape, which we try to localize in the 3D scene using the video images. Localizing and tracking the torso and head of a person in the video feeds is sufficient to localize and track the whole person. Actually, tracking the arms and legs in the video is very complex and will bring no additional interesting information to the detection module. In fact, while walking, these body parts can take different shapes depending on the gait phases. Thus, representing each of these parts with a unique 3D object would be inaccurate. On the other hand, the torso and head 3D forms do not change during the movement and can be represented with a simple invariant 3D object. For all these reasons, we chose to represent the human body with a 3D model of the torso and the head only (Figure 1). As we are trying to estimate this model's 3D position, through an image which is a 2D projection of the real body, our model will be parameterized by its 3D position and orientation (rotation about its vertical axis) in the camera coordinates system. Thus, our model will have four degrees of freedom.

In addition to the 3D human model, we should also develop a function which evaluates its likelihood to the image. In order to compare this virtual model to the real person present in the scene, we compare their 2D respective projections; the 3D model is projected in accordance to its 3D position and orientation, by a virtual camera having the same characteristics as the real camera. The video feed provides the 2D projection of the real 3D scene. We are trying to evaluate the probability of presence of a person in a certain region of the scene, using a 3D model which we position in this region. Thus, in order to compare the synthetic and silhouette images, we only consider the portions of these two images that correspond to this region. This image parts are resized in order to keep a unique scale for all the different configurations sizes (Figure 2). The likelihood w of a certain configuration of the 3D model is calculated by:

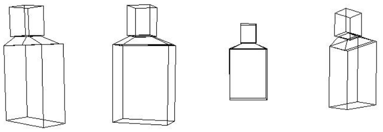
Figure 1. Different configurations of the 3D virtual model used to simulate the human shape. The model is formed by a head and torso represented using cubic volumes. This model is configured through its 3D position and orientation in the scene.

$$w = N_c - (N_s + N_v)$$

where N_c is the number of common pixels in both the synthetic and silhouette images, N_s is the number of pixels of the silhouette image not common with the synthetic image and N_v is the number of pixels of the synthetic image not common with the silhouette image.

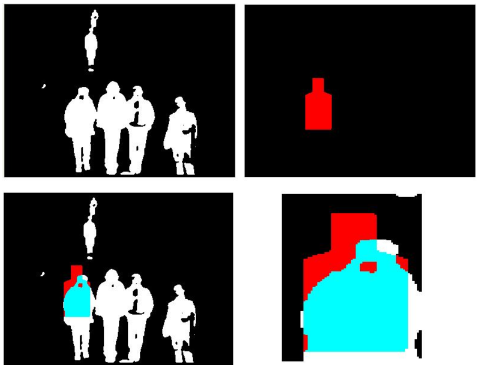
Figure 2. Likelihood evaluation. The silhouette image (top left) is extracted by subtracting the background image and applying a threshold filter and a median filer. One configuration of the 3D virtual model is represented by a synthetic image (top right) in accordance to its position in the 3D scene. The comparison is only applied to the region containing the 3D model (bottom left). This part of the image is then resized (bottom right) and the numbers of common pixels (in blue) and different pixels (in white and red) are counted.

The choice of this function is motivated by the fact that we want to find the configuration which maximizes the number of common pixels and minimizes the number of different ones. This function is simple and can be applied to low resolution images.

4. Explorative PArticle Filtering (ExPF)

The Condensation algorithm also known as Particle Filtering algorithm [7] was designed in order to estimate the *a posteriori* density of a process using the Bayes rule and a set of N samples of the state vector, called particles. At time t the algorithm has a three steps structure:

- Selection (or re-sampling): The set of N particles created at $t-1$ is re-sampled (by N particles) in accordance to the particles weights at $t-1$. Thus, the particles having the greatest weights are selected many times in the new set and those having the smallest weights are vowed to disappear.
- Prediction: Each of the N particles surviving the selection step is updated following a model describing the process dynamics.
- Measure: Given the observation at time t, new weights are assigned to the updated particles. The *a posteriori* density is now represented by these samples and the particle having the greatest weight is considered as the estimation of the state vector. The new weighted N particles set is then used in the selection step at $t+1$.

This algorithm is able to handle multimodal and non Gaussian densities. However, it needs a minimum number of particles to perform a good representation of the density and as a result a precise estimation of the state vector. When the number of used particles is reduced, the number of particles surviving the selection step, also known as survival rate, is reduced. As a result, the few heaviest particles would monopolize the new particles creation and the other configurations would disappear. Thus, the multimodal aspect of the algorithm would be weakened. In this case, a succession of erroneous observations (occlusion for example) will make the algorithm diverge as the most probable estimations are no longer present in the particles set. This problem is known as particles degeneration.

The Interval Particle Filtering [20] introduced for human motion capture modifies the Condensation algorithm in a way to overcome this problem when using a smaller number of particles. In fact, it preserves the advantages of a particle filter algorithm and adopts the same three steps structure with modifications on the selection and prediction steps:

- Selection: Instead of choosing the particles relatively to their weights, a fixed number M of the heaviest distinct particles are selected. By applying this constant survival rate strategy, the particles created are issued from different ones not only from very few when using a small number of particles. The multimodality handling ability is preserved in this way.
- Prediction: No dynamic modelling or white noise is used to update the particles. Using the evolution constraints of each state variable in time, every particle is replaced by a set of particles representing a multi dimensional interval covering the possible configurations of the state vector based on his previous configuration. This approach simplifies the prediction process as no trained dynamic model is used. In addition to the updated particles, a number of fixed particles representing different distinct state vector configurations are added to the set. These static particles guarantee the convergence of the algorithm when all the particles created by selection and update are based on erroneous observations.

The particle filter algorithm can be viewed as a search for the best fitting particle in a set of N created particles representing the state vector search space. The Condensation algorithm uses most of these N particles to intensively populate the most probable zones of this space (established through the previous observation). It tends more to exploit these zones than to explore the entire search space. As a result, when using a reduced number of particles it tracks a single object in a precise way, but acts poorly to detect multiple or newly appearing objects in the scene. On the other hand, the Interval Particle Filtering technique maintains a balance between exploitation and exploration; it uses the N particles in a way to populate more distinct probable zones in a moderate way. Thus, it explores the search space in a better way and offers an ability to detect newly appearing objects. The exploration task is also enhanced in this approach through the introduction of static particles. For these reasons, we opted to use a similar logic for people detection and tracking. The Explorative Particle Filtering (ExPF) we propose here brings some amelioration to the previously described algorithms. Moreover, our approach is conceived in a way to guarantee a good multiple targets tracking.

At time t the three steps of the ExPF algorithm are designed as follows (Figure 3):

- Selection: We use the strategy of fixed survival rate explained previously; from the set of N particles created and weighted at $t-1$, we pick a fixed number of M distinct particles. These M particles represent the M biggest maxima of the sampled a posteriori density $P(X_{t-1}/Z_{t-1})$. This set of M particles is divided into two subsets: The first containing the H particles labelled as representing people at $t-1$ and the second containing B particles considered as representing occluded people or noise. The choice of these particles will be discussed later.

- Prediction: Our 3D model is configured through its 3D position and orientation. As we are tracking walking people, the evolution of these parameters during a single time step (the difference between two images) is constrained by the maximal velocity of human displacement. Thus, a deterministic evolution strategy can be applied to the particles, instead of using a walking trained model or Bayesian noise. In fact, we replace each of the H particles selected earlier by a set of L particles representing a multidimensional interval of particles constructed according to the dynamic constraints of the state variables. These L particles represent the probable current positions of the particle they are replacing. We then add the B particles picked earlier. The remaining $S = N-(H*L+B)$ particles will then be chosen in a way to cover the different regions of the state vector search space. The presence of the B particles and S particles guarantees that the algorithm detects the objects newly appearing in the scene or recovering from an occlusion.

- Measure and decision: The set of N particles created after the prediction step is now weighted given the observation (image) at time t using the likelihood function described in section 3. The *a posteriori* density $P(X_t/Z_t)$ is thus represented by these N weighted samples. By finding the maxima of this function we can find the most probable positions of people in the scene. This can be done by sorting the N particles and picking the M configurations having the greatest weights. But some of the heaviest particles can be very close to each other (having overlapping 2D representation) and thus the M chosen particles would represent some but not all the objects in the scene.

  Our idea is to deal with the N particles as clusters of close particles and not as independent ones. In fact we sort the N particles then classify them into different clusters. The distance between two particles is calculated using the 2D representation of each one; the percentage of the two images' overlapping surface, to the smallest of their respective surfaces would be the distance between the particles. A particle's distance to a cluster is calculated as its distance to the particle considered as its center. When no cluster is found at a minimum distance from a certain particle, a new one is created and the particle would be its center. Else, the particle is assigned to the closest cluster which center will be updated in a way to integrate the new assigned particle. In order to represent the maxima of the *a posteriori* density, we then select the M distinct clusters centers representing the most probable zones instead of the M heaviest particles.

  In order to get precise position estimation, each of these M particles is then optimized regarding the likelihood function. The particle optimization process is similar to a multidimensional gradient descent optimization but without the need to calculate the gradient explicitly. For each of the state variables we consider three possible directions to take: move forward, backward or stay in place. We then combine these possibilities for the three state variables we use to position our model (3D coordinates). Thus, we now have 27 possible directions to take in order to reach the closest maximum weight. For each of these directions we calculate the new weight of the particle (if it moves in this direction), and we move it in the direction which maximizes its weight. We iterate this direction estimation and moving process until no weight increasing direction is found. We then reduce the step with which we move the particle and search again for the direction to take as long as it is possible to maximize the weight. We re-iterate this scheme (estimating-moving then step reduction) until the step we use is small. The orientation is optimized later using the same method.

  The M optimized particles are then weighted and labelled as representing people or noise (occluded person for example) using a threshold. This can be justified by the fact that the likelihood of a person presence in a certain position is equivalent to the weight of the 3D model (particle) which is configured as being at this same position. The H particles labelled as people and B particles labelled as noise constitute the M particles set which will be used in the selection step at $t+1$.

The ExPF algorithm organises the particles search space in an optimal way in order to detect and track multiple targets and has the ability to detect newly appearing objects or those recovering from occlusion. The use of a fixed survival rate as well as the presence of static and noise particles permit a good exploration of the entire space. On the other hand, the optimization process we use increases the estimation precision.

5. Application and results

The 3D model we use is designed in a way to respect a normal body measurments. We run the ExPF algorithm using $N = 625$ particles. For each image we weight the particles and sort them. Using the particles distance described earlier we then classify them into clusters; for each particle we calculate the distances to the already existing clusters. If no cluster was found at a distance smaller than 0.5, a new one is created and the particle is considered as the new cluster's center. Else, the particle is assigned to the closest cluster. When a particle p_j is assigned to a cluster C_i each of its

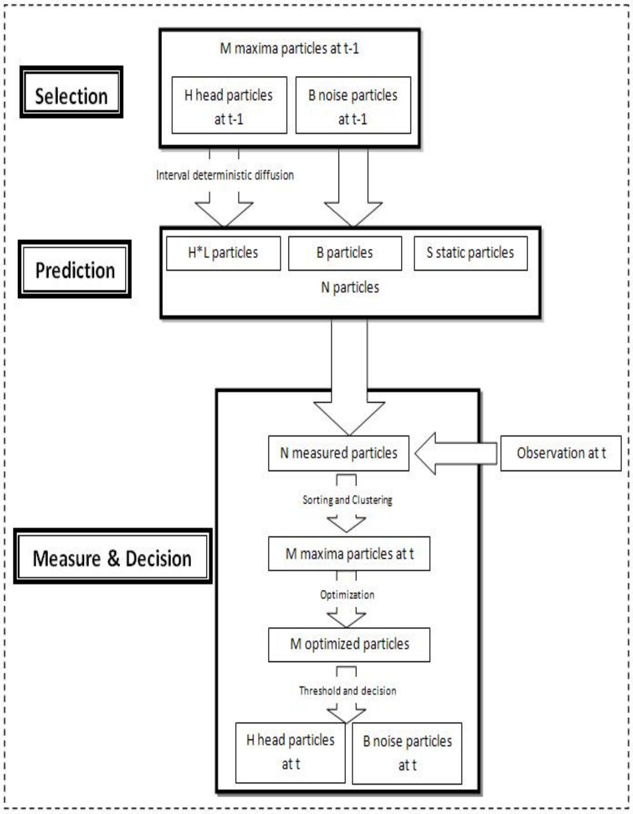

Figure 3. The three steps structure of the ExPF algorithm at t.

center state variables $s^k(k = 1..4)$ is updated as to respect:

$$s^k(C_i) = \frac{\sum_{p_l} s^k(p_l)}{N_i}$$

where p_l is a particle of the cluster C_i and N_i is the number of particles in the cluster C_i.

We consider the $M = 40$ centres of the M clusters containing the heaviest particles as the maxima of the likelihood function. We then optimize these particles as detailed in section 4. These particles can represent people or occluded people or noise (heavy shadows for example). Thus, we use a threshold to discriminate the H *people* particles from others.

We use these M particles in order to construct the N particles set used for detection in the next time step (image). Each of the H particles is replaced by L particles representing the possible configurations of the particle. They are estimated using the current configuration of the particle and the evolution constraints of each of the degrees of freedom of the 3D model. The normal walking speed of a person is about 5Km/h; when dealing with 25Hz video feeds we can estimate the maximal displacement of a person between two frames to be 5cm; If the value of one of the 3D coordinates at time t is a then its value at $t + 1$ will be included in the interval: [a-5;a+5]. This interval can be discretized with 3 values{a-5;a;a+5}; the 3D coordinates of a particle can thus have any configuration of the combinations of these values. As the orientation variation is very difficult to estimate we try to update it during the optimization step only and not through the deterministic update. As a result, we replace each of the H particles with $L = 3^3 = 27$ particles. In addition to these updated particles we add the B particles labeled as noise. The rest of the N particles which we call static particles are chosen as to cover the entire search space.

We applied our approach under these conditions in order to track people in video feeds issued from the CAVIAR benchmark images[1]. We use feeds provided by a single camera at 25images/sec. The ExPF algorithm and methods were developed using C++ and OpenCv functions. The processing was done offline using a P4 3Ghz PC. It takes about 500ms in order to detect and estimate people positions in each frame. The processing time can be reduced by applying a multi threads architecture.

The first set of images (Figure 4) shows the tracking results of two people in the scene with one person disappearing behind a wall and then reappearing in the scene after 2s; the algorithm was able to track this person as soon as it completely reappears in the scene. Our method demonstrates the ability to detect newly appearing objects. That was made possible by the use of the static particles. Actually, all the particles already created around the position of the person before occlusion disappear before the person reappears. In fact, these particles will be related to the background and not to a person anymore, as soon as this person disappears behind a part of this background and thus will have a null weight.

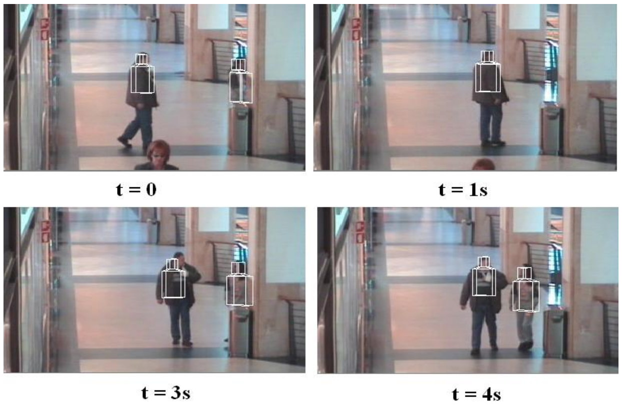

Figure 4. Tracking of people with occlusion by the background. At $t = 0$ (top left image) the person disappears behind the wall and stays entirely occluded by this wall for almost 50 frames (2sec). Thanks to the presence of static particles, the tracker was able to detect the person again as soon as it reappears in the scene (bottom left image). This demonstrates the newly appearing objects detection aspect of our algorithm.

In the second set of images (Figure 5) we try to detect a person briefly disappearing behind another one. These images show that our approach was successful in detecting both people presence even when they are too close to each other and sharing the same blob. In contrary to the first set of images, the person briefly disappears behind another person and not behind a part of the background. The particles created by the person's presence in the scene before being occluded keep having small weights for some consecutive observations as they now describe another person but do not disappear. They are labeled as noise particles. As soon as the occluded person reappears, these particles will have big weights again and thus the algorithm would be able to detect the reappearing body rapidly and precisely. By using the clustering technique we kept these particles alive and we prevented the particles related to the person in front from monopolizing the particles creation.

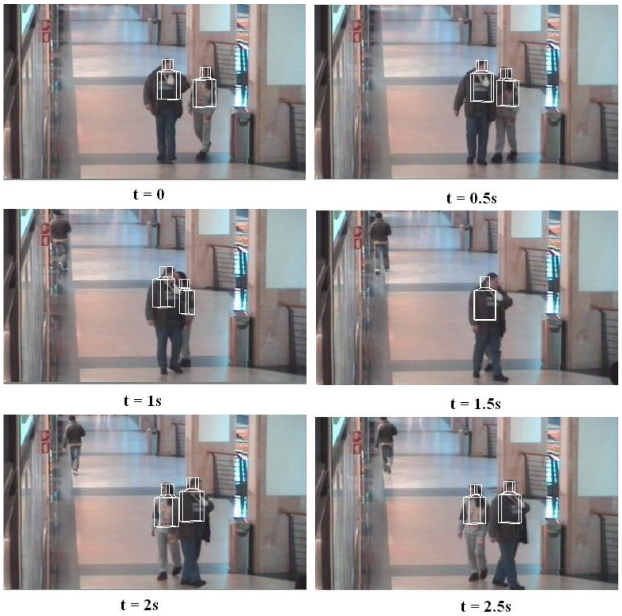

Figure 5. Tracking of people with mutual occlusion. These images show that our method is able to distinguish the presence of both people even when they are close to each other and thus share the same blob (top right and middle left images for example). At $t = 1.5s$ (middle right image) one person is completely occluded by the other. The clustering strategy we adopt keeps the particles related to the occluded person alive but they now have small weights as they evaluate the likelihood of the person in front. They are described as being noise particles. However, as soon as the occlusion was over (bottom left image), they had bigger weights and were labeled as describing a person again.

The last set of images (Figure 6) illustrates the multiple targets aspect of the ExPF algorithm. All the people present in the scene were detected and tracked in a satisfactory way. Thanks to the optimization step, the system was able to give an acceptable estimation of the people positions in the scene even when they are close to each other. As we are using a reduced number of particles, we have a slim probability of directly having the ones that fit the best to the image, in the set of N particles we use. In fact, the heaviest particles in this set were pushed towards the maxima of the likelihood function and thus a more precise estimation was accomplished.

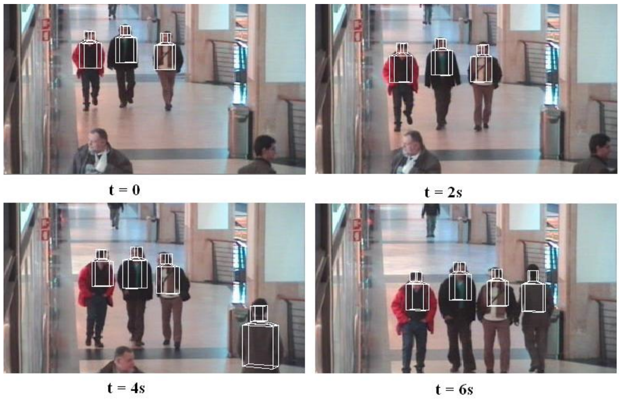

Figure 6. Multiple people tracking. Our system was able to detect and track all the people present in the scene with a good precision thanks to the optimization technique we use. People having only a part of their upperbody in the image are considered as represented by noise particles and thus were not detected.

6. Conclusion

In this article we presented a new method for people detection and 3D tracking in video feeds using a new particle filtering based method. The Explorative Particle Filtering we introduced reorganizes the used particles set in order to have a good exploration of the search space even with a small number of particles. Thus, this algorithm has the ability to track the people who newly enter the scene and those recovering from occlusions. It also provides a good estimation of the 3D positions by optimizing the particles in accordance to the likelihood function applied. We do not use any complex features extraction approach nor trained shapes or colors models. Moreover, we do not apply a dynamic model to update the particles which makes our method simple to apply. The first results are encouraging and demonstrate the tracker's success in detecting people entering the scene or reappearing after being partially or completely occluded. However, some amelioration should be applied to reduce the processing time and enhance the tracking precision especially for the model's orientation. The Explorative Particle Filtering can also be applied in different single or multiple targets tracking tasks and not only for people tracking. It reduces the Condensation algorithm complexity as it needs fewer particles to maintain a balanced exploitation-exploration strategy.

Acknowledgements

This work has been funded by the Ontario Centre of Excellence under the project CR-10030-08 and by the Ontario Research Fund under the project MUSES SECRET ORF-RE2-006 : MUltimodal SurvEillance System for SECurity-RElaTed Research.

References

[1] *The CAVIAR Test Case Scenarios: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.*

[2] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Proceedings of the Intelligent Transportation Systems Conference*, pages 334–339, 2001.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 886–893, 2005.

[4] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proceedings of the 6th European Conference on Computer Vision*, volume 2, pages 37–49, 2000.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W4shydra: Multiple people detection and tracking using silhouettes. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, pages 280–285, 1999.

[6] T. Huang and S. Russell. Object identification: A bayesian analysis with application to traffic surveillance. *Artificial Intelligence Journal*, 103(1-2):77–93, 1998.

[7] M. Isard and A. Blake. Condensation: conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proceedings of the eighth International Conference on Computer Vision*, pages 34–41, 2001.

[9] E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34 (23):93105, 2001.

[10] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Proceedings of the 3rd IEEE Workshop on Visual Surveillance*, pages 3–10, 2000.

[11] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of the IEEE International Conference on Image Processing 2002*, pages 900–903, 2002.

[12] X. Liu, P. H. Tu, J. Rittscher, A. G. A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Proceedings International Conference on Advanced Video and Signal-based Surveillance AVVS05*, pages 306–311, 2005.

[13] W. Lu, K. Okuma, and J. Little. Tracking and recognizing of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1-2):189–205, 2009.

[14] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 572–587, 1999.

[15] A. N. Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo. Automatic estimation of crowd density using texture. *Journal of Safety Science*, 28(3):165–175, 1998.

[16] S. J. Mckenna, Y. Raja, and S. Gong. Tracking color objects using adaptive mixture models. *Image and Vision Computing Journal*, pages 223–229, 1999.

[17] A. Mittal and L. Davis. M2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189203, 2003.

[18] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. People recognition in image sequences by supervised learning. Technical report, MIT AI Memo, 2000.

[19] S. Rao, N. C. Pramod, and C. Paturu. People detection in image and video data. In *Proceeding of the 1st ACM workshop on Vision networks for behavior analysis*, pages 85–92, 2008.

[20] J. Saboune and F. Charpillet. Markerless human motion tracking from a single camera using interval particle filtering. *International Journal on Artificial Intelligence Tools*, 16(4):593–609, 2007.

[21] H. Tsutsui, J. Miura, and Y. Shirai. Optical flow-based person tracking by multiple camera. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 91–96, 2001.

[22] T. Tung and T. Matsuyama. Human motion tracking using a color-based particle filter driven by optical flow. In *ECCV workshop on Machine Learning for Vision-based Motion Analysis ECCV'08*, 2008.

[23] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):1573–1405, 2004.

[24] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of the International Conference on Computer Vision ICCV2003*, pages 734–741, 2003.

[25] C. Wen, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real time tracking of human body. *IEEE Transactions on PAMI*, 19(7):780–785, 1997.

[26] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proceedings of the International Conference on Computer Vision ICCV2005*, pages 90–97, 2005.

[27] J. Xu, G. Ye, G. Herman, and B. Zhang. An efficient approach to detecting pedestrians in video. In *Proceedings of the ACM Multimedia Conference 2008*, pages 789–792, 2008.

[28] Z. Zeng and S. Ma. Head tracking by active particle filtering. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 82–87, 2002.

[29] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004*, pages 1208–1221, 2004.