OCCLUSION-HANDLING FOR IMPROVED PARTICLE FILTERING-BASED TRACKING

Raphaël Canals, Ali Ganoun, and Rémy Leconge

PRISME Institute, University of Orléans
12 rue de Blois, BP6744, 45067 Orléans cedex 2, France
phone: + (33) 2 38 49 45 58, fax: + (33) 2 38 41 72 45, email: raphael.canals@univ-orleans.fr
web: www.univ-orleans.fr/prisme

ABSTRACT

One of the particle filtering uses is object tracking since this technique permits to deal with uncertainty over time met in real time image sequences framework. This uncertainty is as much non-manageable that an object occlusion appears in images. In this paper, we propose an occlusion-handling scheme which significantly improves the tracking performance in presence of partial occlusion. The proposed technique is applied to track a single object in real greyscale image sequences. Results confirm tracking performance enhancement.

1. INTRODUCTION

Object tracking techniques aim at following objects in image sequences. They should be able to deal with complex interactions and various dynamics in sequences such as occlusions, camera motion, varying lighting conditions and viewing directions. Object tracking is useful in many image-based applications including video communication/compression [1] and surveillance systems [2].

Filtering and data association techniques are widely applied in computer vision for various tracking applications, as in the work of Rasmussen and Hager [3] who adapt probabilistic data association filters and joint probabilistic data association filters for tracking complex visual objects. Within the filtering and data association approach, the particle filter technique will be particularly concerned here: it is a Bayesian methodology which applies a recursive filter, based on samples of the object to be tracked [4], [5].

This paper considers the particle filter technique. While the particle filter is usually used with colour sequences, we chose to use the grey-level scale because of its lower data size with a view to implementing it in real time on an embedded system. Moreover some application domains such as video surveillance in which we are interested implement still grey-level cameras. We also chose a context without any a priori information in order to be closer to real working conditions. Consequently, we cannot use any learning phases: we only have a single model of the object to be tracked extracted from the first image.

Another problem related to object tracking is that of occlusion, whether it is partial or complete. Partial occlusion hides some parts of the target while complete occlusion hides the entire target for some time. Many techniques exist to handle the occlusion problem with particle filter probabilistic models, such as in [6], [7],[8] and in the work of Nummiaro and al.[9] who present a system to track objects in presence of occlusion. The proposed tracking method adds the robustness and invariance of colour distributions to particle filtering. The probabilistic tracking model proposed in [10] uses a particle filter for a better handling of colour clutter in the background, as well as complete occlusion of the tracked object over a few frames. Jepson and al. [11] propose an adaptive recursive approach by employing a mixture of three appearance probabilistic components: a stable component, a two frame transient component, and an occlusion component to deal with outliers. In [12] and [13], authors have proposed complex particle tracking algorithms in colour sequences associated with a tracker uncertainty evaluation.

In this paper, we propose an occlusion-handling scheme based on particle filter framework. Zhou et al [14] have already proposed a similar technique; in their paper, the visual tracker relies on an adaptive appearance model, a velocity motion model with adaptive noise variance, and an adaptive number of particles, with occlusion handling via robust statistics. The occlusion is declared when the number of outliers in the object of interest compared with the appearance model exceeds a threshold: therefore the appearance model must not be updated. Our approach differs from their solution in that it does not keep solely affine transformations and in the technique of considering the occlusion, as explained below in greater detail.

The rest of the article is structured as follows: in the next section, we present the principle of particle filtering. Our particle filter version is proposed in Section 3. Section 4 demonstrates the results of the proposed approach using several real scene sequences. The last section terminates this paper by concluding on our work.

2. PATICLE FILTERING

Particle filtering (PF) is a sophisticated method derived from the Bayesian recursive filtering for model state estimation; it is a promising technique as it models uncertainty and can, with sufficient samples, deal with many tracking problems such as missing data and occlusions. It is known under different names including the Monte Carlo approach [15], the CONDENSATION algorithm [5] and bootstrap filter [16]. One of the main properties of the particle filter is that it gives an approximate solution to an exact model, rather than the optimal solution to an approximate model as with Kalman filters. It handles non linear models with non-Gaussian noise; as a result, it has been proven to be a powerful technique for tracking non linear systems.

The basic idea of this technique is to evaluate the position of an object by testing its presence on a limited number of points. When this principle is used on object tracking, the result is a local similarity test between the target model and the image, done for every pixel [5]. The output of this type of tracking is not an absolute value. In our case, the response is a bi-dimensional map indicating the probability of locating the object in the picture, i.e. the probability density is approximated by a set of weighted particles.

The first step of the tracking algorithm is the initialization in which the target is detected and defined; a random number of particles are uniformly distributed inside the target in order to represent it correctly. Each particle is represented by its state vector X_k, $k \equiv \{1 \ldots N\}$ where N is the number of particles. The initial state vector is given as:

$$X^k = \begin{pmatrix} x & y & v_x & v_y \end{pmatrix}^T \tag{1}$$

where x and v_x are the position and speed in x direction respectively, and y and v_y in the y direction. Initially their respective speeds are null.

The second step is the prediction step, where each particle is modified according to the state model of the region of interest in the video frame. This prediction corresponds to a propagation of particles X^k at time t-1 and is given by:

$$X_t^k = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .X_{t-1}^k + \eta_t^k$$

η_t^k is the noise of the transition model. Around each particle considered as a test point, the target model corresponding to its grey-level distribution is compared with the local grey-level distribution. This comparison is carried out by using the Bhattacharyya coefficient although it has been studied in [17] to give biased results with greyscale images as there are not enough information in the histogram. As the main goal of this application is to track a target and not to define exactly the target, the use of this coefficient derives from a compromise between accuracy and lightness of the implementation; this coefficient is defined as:

$$z_t^k = \sum_{l=0}^{L} \sqrt{p^k(l) \times q^k(l)} \tag{2}$$

where z_t^k is a similarity criterion, L is the number of values which can be taken by each pixel (256 for a standard grey-scale image), p^k is the model histogram and q^k that of a local area around the k^{th} particle. This step is called the weighting or update step.

As expressed in [16], one simple way to obtain better results is to use the exponential of the criterion distance:

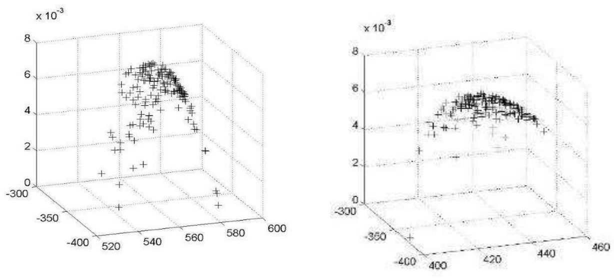
$$w_t^k = \exp\left(-\left(1\text{-}z_t^k\right)\right) \Big/ \sum_{k=1}^{N} z_t^k \tag{3}$$

This criterion is often used in colour-based pictures. Applied to grey-scale images, it requires more information to be relevant. To meet this need, a spatial dimension can be used, for example a bi-dimensional weighting window which permits to weight the pixels by considering theirs distances to the window centre [18].

Many factors, such as the number of particles, the appearance model, and the particle motion model, affect the tracking result. The global result of the tracking is given by the mean state of the particles, corresponding to the state estimate deriving from the particle approximation of the posterior probability, i.e.:

$$\bar{X}_t = \sum_{k=1}^{N} X_t^k w_t^k \tag{4}$$

The last step is the resampling procedure which eliminates particles that have small weights, i.e. low probability, and replicates the particles with larger weights, i.e. high probability, in the target. This procedure consists in fact in a particle redistribution which preserves only the most reliable ones.



Good detection Occlusion or bad detection

Figure 1 – Effect of an occlusion on particles.

3. THE MODIFIED TECHNIQUE

The results of the standard technique [5] is acceptable with some simple sequences. However, the algorithm fails to track the target in complex sequences in which there are some occlusions or higher appearance changes of the object to be tracked. In this section, we present the modification made to the standard technique in order to improve the tracking algorithm in such cases.

The principle of the occlusion detector we have developed is obtained from the observation of the weighting operation result. In the event of occlusion or bad detection, the bi-dimensional similarity function is levelled, as shown in Figure 1. This means that the similarity maximum is more difficult to detect and more sensitive to noise: it is more complicated to define where the object is located.

We therefore propose to evaluate the flatness of the result by using a dispersion criterion:

$$D_t = \frac{\sum_{k=1}^{N} w_t^k \times \left|X_t^k - \bar{X}_t\right|^2}{Sx^2 + Sy^2} \tag{5}$$

$\bar{X}_t$ is the mean state vector of the particles, Sx and Sy are the model dimensions on the x-axis and y-axis respectively.

The second recurrent problem is target deformations due to the relative displacement of the target and the camera or simply to natural target deformations. It is therefore not possible to keep the same target size and form throughout the sequence and it is necessary to employ a deformable model in order to manage this problem [19].

The effect of the resample step is to gather the particles around the position where the presence probability is higher. After this step, particles are closer to the object, and they are, in most cases, inside it. So, a very simple way to evaluate the model size and its topology is to use the position of these particles after the resample step: a morphological closing operation is applied in order to define the object (Figure 2). The resampling depends on the effective sample size referring to the number of particles expected to survive from this step; it is defined as:

$$N_{eff} = 1 \Big/ \sum_{k=1}^{N} \left(w^k\right)^2 \tag{6}$$

The inefficient particles ($\bar{N}_{eff} = N - N_{eff}$, ordered by weight) are redistributed around the central target position. This distribution goes by the normal law. The distribution centre is the initial target one and its deviation is fixed to the third of the target size. Theory indicates that the use of a Gaussian distribution of ±3 σ gives us a target covering of 99%.

Particular care must be taken in case of trouble during this step by testing the similarity between the pixels included in the mask and

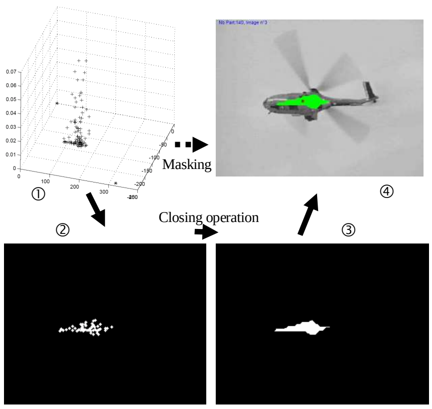


Figure 2 – Masking step description.

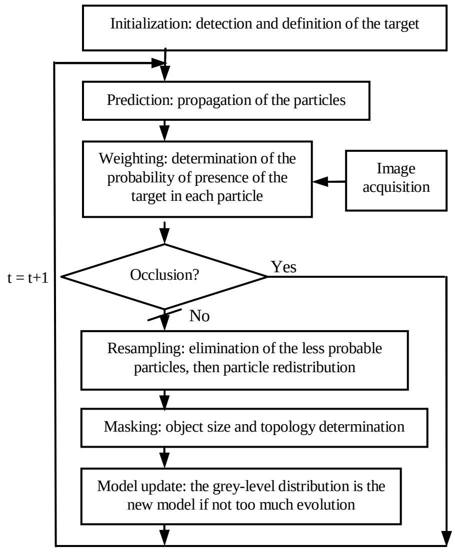
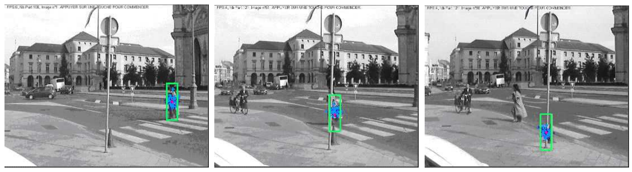


Figure 3 – Improvement of the basic algorithm.

the model previously defined, to avoid a region belonging to the background being considered as a part of the target.
Because of the closing operation during the masking step, the target topology is limited: the target must not include holes or transparent parts with background grey-level distribution. To preserve satisfactory operating conditions, an update step is performed only if the grey-level distribution in the mask is close enough to the target model. In this case, the grey-level model distribution becomes the distribution of the area included in the mask; otherwise the grey-level model distribution remains unchanged.
The final tracking algorithm is given in Figure 3. Compared to the standard algorithm [5], we can note that occlusion detection and the masking technique have been added for improving the tracking.

4. RESULTS

The modified algorithm was tested on numerous greyscale image sequences, with a target initialisation realized thanks to a specific region growing algorithm. Only three representative sequences are presented here.
The first sequence is the OC2 sequence in which we are interested in tracking a woman. This sequence is composed of 97 images of 720x576 pixels. The appearance of the target does not change a much, but it is partially occluded during approximately 10 images.
The second sequence is the OC1 one of 175 images (720x576 pixel) in which the objective is to track a pedestrian who is occulted suc-



Frame 1 *Frame 61* *Frame 90*

Figure 4 – Tracking results of the standard algorithm on the OC2 sequence.

Frame 60 *Frame 62* *Frame 64*

Figure 5 – Tracking results of the improved algorithm on the OC2 sequence.

cessively by a cyclist and another pedestrian.
The third sequence is the Univ sequence which is one of the most complex ones amongst the test sequences. Indeed not only the sequence is very noisy but the pedestrian who constitutes the target undergoes also very large occlusions in addition to the simultaneous displacement of the target and the camera.
One of the main PF configuration parameters is the particle number influencing the processing time. This number is also a factor of precision on the probability of target presence.
It is important to satisfy these two points: processing time and precision. When the particle number rises, processing time increases too and the speed of the target averaged on the ten last images reduces because of a better target localisation with a greater precision. A lack of particles produces an effect of oscillation around the real target position and creates a dispersion of the estimated position. The particle number must then be chosen with attention. Several tests on many image sequences have demonstrated that this number can be defined as a target size percentage in the first image: a value of 10% gives good results.
One of the biggest problems in standard PF tracking is to track an object through occlusion (Figure 4 in which the box surrounds the maximal limits of the object to be tracked): the PF tracker converges to a local maximum in the background. In contrast, our PF algorithm tracks successfully the target (Figure 5).
The ratio of the dispersion coefficient to its mean, for the OC2 sequence, is presented in Figure 6. Because the target environment is changing, we use the mean of the previous dispersion function values to set the detection threshold. The first falling edge is caused by the incomplete average operation. In this way, occlusion detection can be easily determined with a simple threshold. To eliminate any ambiguous detection, a trigger must be used. So there is detection-when the dispersion function rises above the triple of the mean of the previous values. Detection ends when it falls above that mean.
To evaluate the tracking stability, the algorithm is repeated 10 times with each sequence, with a different initialization of the target. In fact, it is assumed that the target is located within a given window.
Therefore, the operator has to realize the target selection at the beginning of each trial by defining a window surrounding the object to be tracked. The performance of the tracking algorithm is estimated by calculating the tracking error and the percent of convergence.
The tracking error, expressed in number of pixels, represents the deviation of the algorithm result from the reference result (Ground Truth). The Ground Truth is the ideal representation of a target over

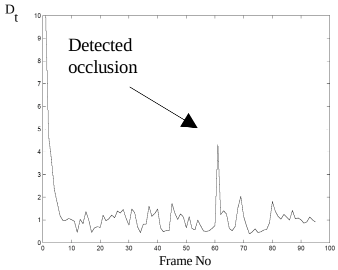


Figure 6 – Variation of the dispersion function of OC2.

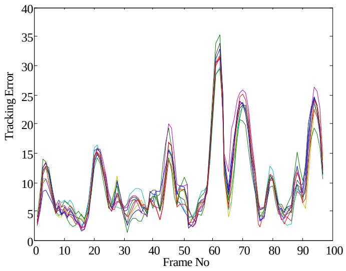


Figure 7 – Performance curves with our algorithm on the OC2 sequence.

time and is defined manually. The percent of convergence corresponds to the percent of trials for which the algorithm gives an accurate result. The performance curves represent the tracking error with the time [20], [21].

The performance curves for some selected experiments on the OC2 sequence are shown in Figure 7, the figure indicating the tracking error at each frame for each trial. The results are nearly identical for all the trials. The percent of convergence with this sequence is 100% though the tracking error fluctuates along the sequence, with a peak during the occlusion. Despite this error, our algorithm permits to track but not to locate precisely the pedestrian, which is its objective.

An example of target tracking failure with the standard algorithm on the sequence OC1 is given in Figure 8. The corresponding performance curves show that the algorithm fails when there are high occlusions near the frame 68. In some trials, the algorithm even fails from the occlusion with a cyclist near the frame 37. This sequence is enough complex with similar pixel grey-levels between the pedestrian and the background, and some occlusions difficult to manage.

For the same sequence, some results of target tracking with our algorithm are shown in Figure 9. The performance curves are given in Figure 10. From the example and performance curves, it can be noted that the algorithm tracks successively the target, in spite of the tracking errors, mainly during the large occlusions and because of similar grey-levels regions in the background. It is to be noted that depending on the particle propagation, the tracking result can change, as represented near the frame 72 in Figure 10.

For the same sequence, some results of target tracking with our algorithm are shown in Figure 9. The performance curves are given in Figure 10. From the example and performance curves, it can be noted that the algorithm tracks successively the target, in spite of the tracking errors, mainly during the large occlusions and because of similar grey-levels regions in the background. It is to be noted that depending on the particle propagation, the tracking result can change, as represented near the frame 72 in Figure 10.

Some results of applying the PF algorithm on the Univ sequence shown in Figure 11 illustrate the difficulty of tracking the target in this sequence. One time out of two, the percent of convergence is zero (Figure 13), depending on the initialization and the particle

Figure 8 – Example of target tracking failure with the standard algorithm on the OC1 sequence.

Figure 9 – Example of target tracking success with our algorithm on the OC1 sequence.

distribution. One tracking failure is shown in Figure 15, and we can notice the difficulty of the pedestrian tracking. As in the previous sequence, some pedestrian pixels have similar grey-levels to some ones of the background, and the two occlusions are hard to manage. Finally one tracking success is presented in Figure 12.

5. CONCLUSION

We have presented an improved particle filter tracking algorithm suitable for object tracking in video sequences. The new approach is very robust as it can overcome occlusion of the tracked object, as well as tracking noise such as varying lighting conditions and viewing directions. On the other hand, if there is a large occlusion and a part of the background is similar to the target, the algorithm can fail because it is focused on the background and does not return on the target at the end of occlusion.

This algorithm is quite easy to implement and not time-consuming. It could be used for many applications in which the target cannot be

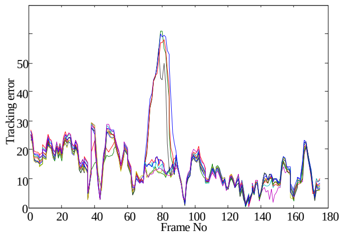


Figure 10 – Performance curves with our algorithm on the OC1 sequence.

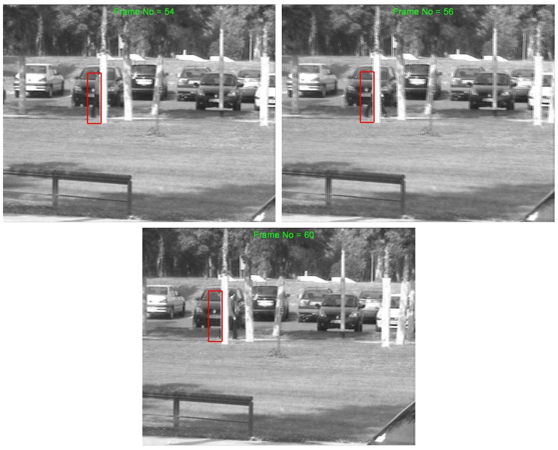


Figure 11 – Failure of tracking with our PF algorithm on the Univ sequence.

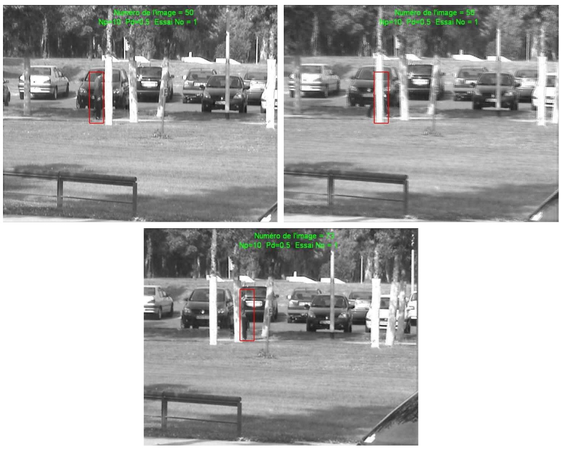


Figure 12 – Success of tracking with our PF algorithm on the Univ sequence.

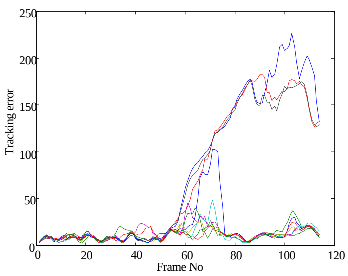


Figure 13 – Performance curves with our algorithm on the Univ sequence.

totally extracted: the pan / tilt camera control is a perfect example of use. It could also be implemented to track several objects; in this case, if the objects are similar, the dispersion function must be changed to avoid any merging effect.

REFERENCES

[1] Menser, B. and M. Brunig. Face detection and tracking for video coding applications. In 34th Asilomar Conf. on Signals, Systems and Computers. 2000. Pacific Grove, California, USA.
[2] Greiffenhagen, M., et al. Statistical modeling and performance characterization of a real-time dual camera surveillance system. in Computer Vision and Pattern Recognition. 2000.
[3] Rasmussen, C. and G. Hager, Probabilistic Data Association Methods for Tracking Complex Visual Objects. IEEE Trans. on Pattern Anal. and Machine Intelligence, 2001. 23(6): p. 560-576.
[4] Gordon, N., D. Salmond, and C. Ewing, Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter. Journal of Guidance, Control and Dynamics 1995. 18(6): p. 1434-1443.
[5] Isard, M. and A. Blake, CONDENSATION - Conditional Density Propagation for Visual Tracking. International Journal of Computer Vision, 1998. 29(1): p. 5-28.
[6] Khan, Z., T. Balch, and F. Dellaert. MCMC-based Particle Filtering for Tracking a Variable Number of Interacting Targets. IEEE Trans. on PAMI. 2005, vol.27, N°11, p.1805-1819.
[7] Qu, W., et al. Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model. IEEE Trans. on Multimedia. 2007, vol. 9, no3, p. 511-519.
[8] Czyz, J., B. Pistic, and B. Macq. A particle filter for joint detection and tracking of colored objects. Image &Vision Computing. 2007, vol. 25, issue 8, p. 1271-1281.
[9] Nummiaro, K., and al., A Color-Based Particle Filter. in First International Workshop on Generative-Model-Based Vision. 2002.
[10] Perez, P., et al. Color-based Probabilistic Tracking. in European Conference on Computer Vision. 2002.
[11] Jepson, A., D. Fleet, and T. El-Maraghi, Robust Online Appearance Models for Visual Tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003. 25(10): p. 1296-1311.
[12] Maggio, E., et al. Combining Colour and Orientation for Adaptive Particle Filter-based Tracking. in 16th BMVC. 2005.
[13] Badrinarayana, V., et al., Probabilistic Color and Adaptive Multi-feature Tracking with Dynamically Switched Priority between Cues. in ICCV, Oct. 2007, p.1-8.
[14] Zhou, S., R. Chellappa, and B. Moghaddam. Appearance Tracking Using Adaptive Models In a Particle Filter. in Asian Conference on Computer Vision. 2004.
[15] Kitagawa, G., Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. Journal of Computational and Graphical Statistics, 1996. 5(1): p. 1-25.
[16] Gordon, N.J., D.J. Salmond, and A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. on Radar and Signal Processing, 1993. 140(2): p. 107-113.
[17] Sohail, K., et al., Bhattacharyya Coefficient in Correlation of Grey-Scale Objects Journal of Multimedia, 2006. 1(1): p. 56-61.
[18] Oppenheim, A. and R. Schafer, Discrete-time signal processing. Prentice-Hall Signal Proc. 1989, Eng. Cliffs, Prentice Hall.
[19] Nummiaro, K., et al. Object Tracking with an Adaptive Color-Based Particle Filter. in Symp. for Pat. Rec. of the DAGM. 2002.
[20] Erdem, C., B. Sankur, and A. Tekalp, Performance measures for video object segmentation and tracking. IEEE Transactions on Image Processing, 2004. 13(7): p. 937-951.
[21] Black, J., et al., A Novel Method for Video Tracking Performance Evaluation. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2003. France.