Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 53(67), Fascicola 2, 2008

Occlusions Detection for Improved Particle Filtering-based Tracking

Raphael Canals, Ali Ganoun [1]

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.

Keywords: object tracking, particle filtering, occlusion

I. 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. 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 the work of Nummiaro and al. [6] who present a system to track people 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 [7] 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. [8] 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 this paper, we propose an occlusion-handling scheme based on particle filter framework. Zhou et al [9] 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.

II. PARTICLE FILTERING

Particle filtering (PF) is a sophisticated method 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

[1] University of Orleans, PRISME Institut
12 rue de Blois, BP6744, 45067 Orleans cedex 2, France, e-mail raphael.canals@univ-orleans.fr

data and occlusions. It is known under different names including the Monte Carlo approach [10], the CONDENSATION algorithm [5] and bootstrap filter [11]. 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], [12]. 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 CONDENSATION 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, $k \in \{1 \ldots N\}$ 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 a noise vector used to simulate the noise in the state vector, and to explore the expected presence probability during the sequence.

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 [13] 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.

One simple way to obtain better results is to use the exponential of the criterion distance:

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

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 [14].

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, i.e.:

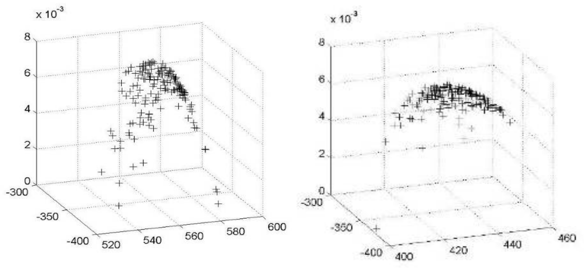
$$\bar{X}_t = \sum_{k=1}^{N} X_t^k w_t^k \tag{4}$$

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.

III. 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 the Fig.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.



Good detection Occlusion or bad detection

Fig.1. Effect of an occlusion on particles.

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 position 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 [15].

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 (Fig.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 the model previously defined, to avoid a region belonging to the background being considered as a part of the target.

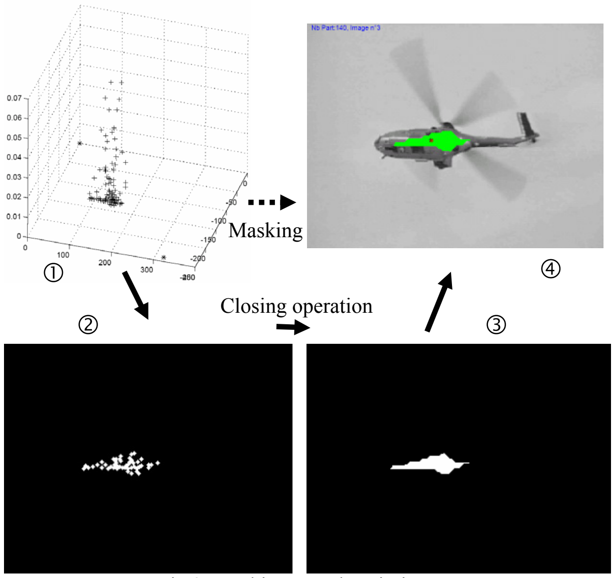


Fig.2. Masking step description.

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 Fig.3. Compared to the standard algorithm [5], we can note that occlusion detection and the masking technique have been added in order to improve the tracking.

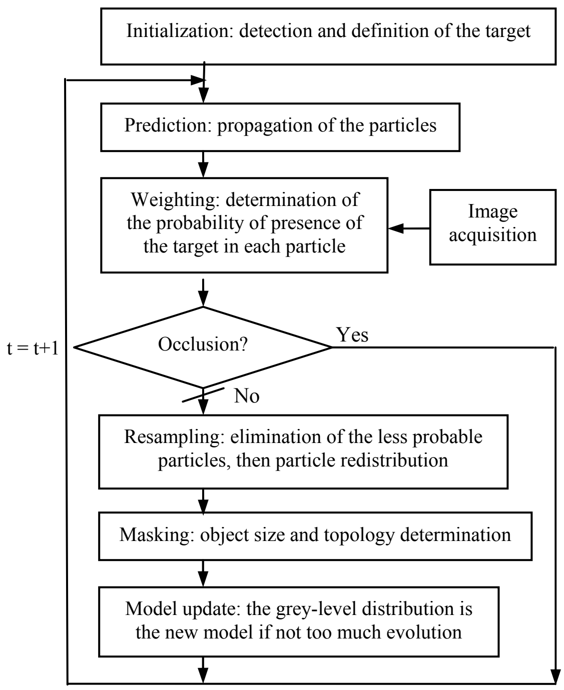


Fig.3. Improvement of the basic algorithm.

IV. RESULTS

The modified algorithm was tested on numerous greyscale image sequences. Only four 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.

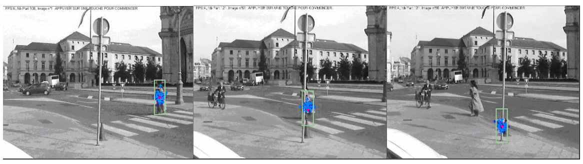
HC1 is the second sequence of 244 320x240-pixels images in which the appearance of the target changes: the helicopter is often partially hidden by its own blades.

The third sequence is the Road one of 90 images (320x240 pixels) in which the objective is to track a vehicle. The main characteristic of this sequence is the target size changes through the time.

The OC2 sequence concerns the tracking of a pedestrian which is occulted by a cyclist. This sequence contains 175 images of 720x576 pixels.

One of the biggest problems in standard PF tracking is to track an object through occlusion (Fig.4): the PF tracker converges to a local maximum in the background. In contrast, our PF tracks successfully the target (Fig.5).

The ratio of the dispersion coefficient to its mean, for the OC2 sequence, is presented in Fig.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.



Frame 1 *Frame 61* *Frame 90*

Fig.4. Tracking results of the standard algorithm on the *OC2* sequence.

Frame 60 *Frame 62* *Frame 64*

Fig.5. Tracking results of the improved algorithm on the *OC2* sequence.

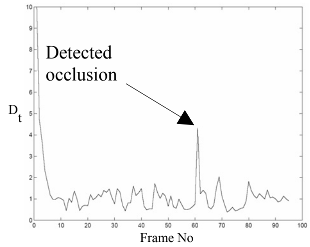


Fig.6. Variation of the dispersion function of *OC2*.

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 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 [16], [17], [18].

The performance curves for some selected experiments on the OC2 sequence are shown in Fig.7, the figure indicating the tracking error at each frame for each trial. The percent of convergence with this sequence is 100%.

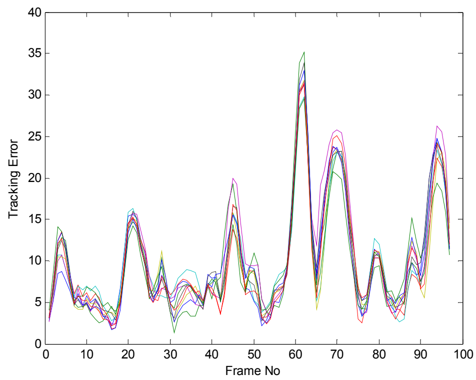


Fig.7. Performance curves with our algorithm on the *OC2* sequence.

For the HC1 sequence, the performance curves are shown in Fig.8. Our algorithm tracks easily the helicopter through all the frames; indeed the target is well distinguished from the background. The results are nearly identical for all the trials. From the performance curves, we note the highest error when the target changes its form. The tracking success can be seen in the examples of the Fig.9. As with the OC2 sequence, the percent of convergence for this sequence is 100%.

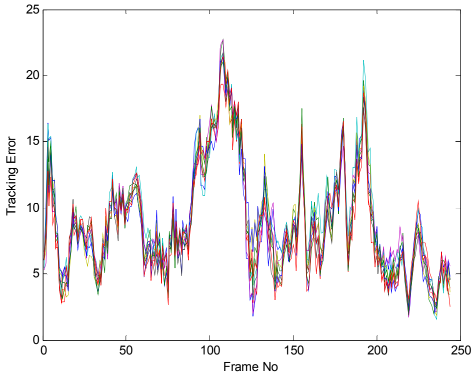


Fig.8. Performance curves with our algorithm on the *HC1* sequence.

An example of target tracking in the Road sequence is shown in Fig.10. The performance curves are given in Fig.11. From the example and performance curves, it can be noted that the algorithm tracks successively the target, in the beginning, but as the target size decreases, it fails to track it. The algorithm performance deteriorates from about the frame 40. In this case, the percent of convergence is 0%.

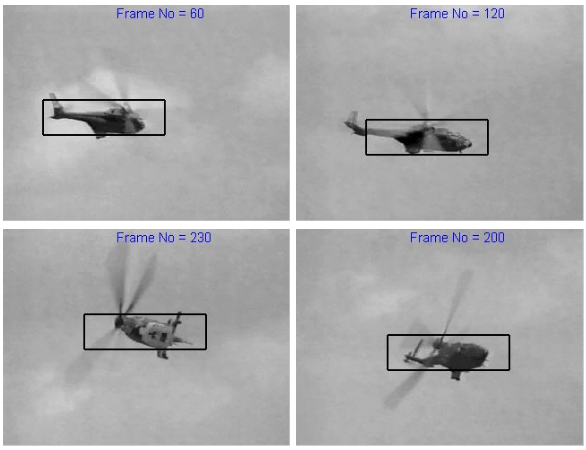


Fig.9. Examples of target tracking with our algorithm on the *HC1* sequence.

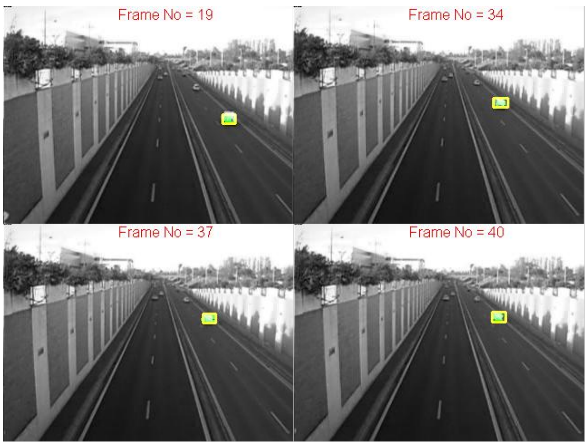


Fig.10. Examples of target tracking with our algorithm on the *Road* sequence.

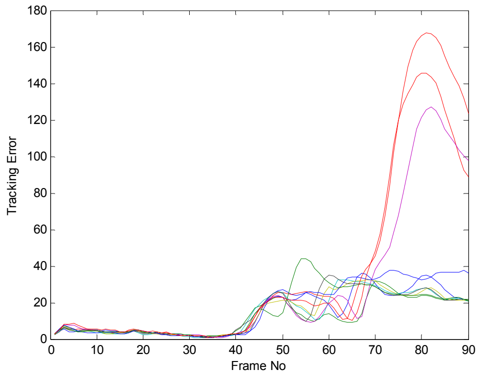


Fig.11. Performance curves with our algorithm on the *Road* sequence.

An example of result on the sequence OC1 is given in Fig.12. Fig.13 indicates the failure of our FP to track the pedestrian in this sequence; the performance curves show that the algorithm fails when there are high occlusions near the frame 70. In some trials, the algorithm even fails from the occlusion with a cyclist near the frame 52. This sequence is enough complex with similar pixel grey levels between the pedestrian and the background, and some occlusions difficult to manage.

Fig.12. Examples of target tracking with our algorithm on the *OC1* sequence.

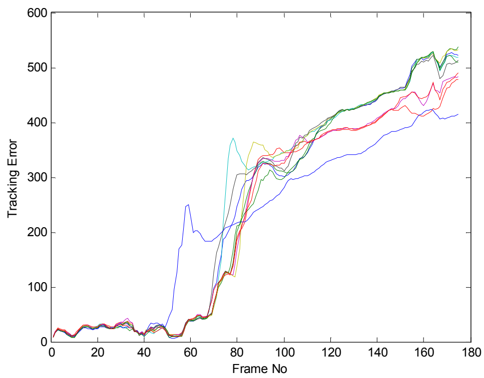


Fig.13. Performance curves with our algorithm on the *OC1* sequence.

V. 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 fails 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 is not time-consuming. It could be used for many applications in which the target cannot be 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 Conference 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 Transactions on Pattern Analysis 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] Nummiaro, K., E. Koller-Meier, and L. Van Gool. A Color-Based Particle Filter. in First International Workshop on Generative-Model-Based Vision. 2002.

[7] Perez, P., et al. Color-based Probabilistic Tracking. in European Conference on Computer Vision. 2002.

[8] Jepson, A., D. Fleet, and T. El-Maraghi, Robust Online Appearance Models for Visual Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. 25(10): p. 1296-1311.

[9] Zhou, S., R. Chellappa, and B. Moghaddam. Appearance Tracking Using Adaptive Models In a Particle Filter. in Asian Conference on Computer Vision. 2004.

[10] 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.

[11] Gordon, N.J., D.J. Salmond, and A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings on Radar and Signal Processing, 1993. 140(2): p. 107-113.

[12] Jacquot, A., P. Sturm, and O. Ruch. Adaptative Tracking of Non-Rigid Object Based on Color Histograms and Automatic Parameter Selection. in ORASIS. 2001.

[13] Sohail, K., et al., Bhattacharyya Coefficient in Correlation of Grey-Scale Objects Journal of Multimedia, 2006. 1(1): p. 56-61.

[14] Oppenheim, A. and R. Schafer, Discrete-time signal processing. Prentice-Hall signal processing series. 1989, Englewood Cliffs, N.J.: Prentice Hall.

[15] Nummiaro, K., E. Koller-Meier, and L. Van Gool. Object Tracking with an Adaptive Color-Based Particle Filter. in Symposium for Pattern Recognition of the DAGM. 2002.

[16] 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.

[17] Black, J., T. Ellis, and P. Rosin. A Novel Method for Video Tracking Performance Evaluation. in International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2003. Nice, France.

[18] Cavallaro, A. and F. Ziliani. Characterisation of tracking performance. in 6th International Workshop on Image Analysis for Multimedia Interactive Services. 2005. Montreux, Switzerland.