2012 IEEE 24th International Conference on Tools with Artificial Intelligence

# Visual Tracking by Weighted Likelihood Maximization

Vasileios Karavasilis, Christophoros Nikou and Aristidis Likas
*Department of Computer Science, University of Ioannina*
*45110 Ioannina, Greece*
{vkaravas,cnikou,arly}@cs.uoi.gr

***Abstract*—A probabilistic real time tracking algorithm is proposed. The distribution of the target is represented by a Gaussian mixture model (GMM) and the weighted likelihood of the target is maximized in order to localize it in an image sequence. The role of the weight is important as it allows gradient based optimization to be performed, which would not be feasible in a context of standard likelihood representations. The algorithm models both the object to be tracked and local background elements and handles scale changes in target's appearance. It is experimentally demonstrated that the algorithm runs in real time, and it is at least at the same performance level with the mean shift algorithm while it provides more accurate target localization in non trivial scenarios (e.g. shadows).**

## I. Introduction

Visual tracking is the process of locating an object's position in the frames of an image sequence. As described in [1], the trajectory of the object over time can be obtained either a) by estimating the object's position in a frame based on the position in the previous frame or b) by detecting the object in every frame and afterwards to associate the detections between them. The algorithms of the first group usually deal with one object whose location and appearance obtained from the previous frame are updated based on the observations of the current frame. The second group usually handles many targets and the objective is both to separate them in a single frame and find their correlation between frames. These groups of algorithms can also be further subdivided into categories, depending on the model that is used to represent the target.

The simplest representation of an object is a vector which defines the state. The state may be a combination of the location, the velocity and the acceleration. Methods that rely on filtering combine a prediction made from previous states and an observation generated from the current frame to estimate the current state. These methods include Kalman filter [2] and particle filters [3] and they can estimate the object's position even if no observations are provided (e.g. due to occlusions) but they assume a state evolution model which must be defined accurately in advance.

Another category of methods assume that the object has a relatively simple shape (e.g. an ellipse or a rectangle) which is spatially masked with a kernel. These methods usually rely only on the observations to estimate the object's position. They start from an initial location (which is usually the position of the object in the previous frame) and use a gradient-decent based optimization procedure to estimate the objects position. This optimization is performed through a cost function which is usually a distance between histograms, such as in the mean shift [4] and the Camshift [5] algorithms, histogram signatures [6] or Gaussian mixture models [7]. In [8] the author shows that the mean shift is an EM algorithm if the kernel is gaussian and a generalized EM if a non-gaussian kernel is used and in [9] the mean shift is treaded as an EM-like algorithm in order to estimate the orientation of the target in addition to the position and the scale. One drawback of these methods is that they can not handle total occlusions. Methods that combine these techniques with algorithms such as Kalman filter [10] and particle filters [11] have also been proposed in order to overcome their limitations.

A more detailed representation of the shape of the target can be achieved through level sets or active contours, which were successfully used in tracking [12]. This representation was employed to track multiple objects [13], [14] along with its combinations with other approaches, such as particle filters [15]. In [16], active contours are combined with Bayesian filters to robustly segment the object from the background.

The above methods assume an appearance model that is initialized in the first frame and tracked in consecutive frames. If the appearance of the object changes, the appearance model must be updated too. Mixture models have been combined with tracking algorithms in order to update the appearance model in cases where the object is represented by histograms [17] or level sets [18]. In [19], multiple instance learning is used to update the appearance model in cases of partial occlusion.

The majority of these algorithms deal with one object. Although a distinct tracker can be used in order to track many objects simultaneously, this is not the optimal solution as these objects may partially or totally occlude each other and this information is not handled by the tracker. Therefore, more advanced algorithms have been designed in this framework. In [20], graph cuts were used in order to segment the frame into possible objects and associate them with objects detected in previous frames. In [21], multiple objects are also

V. Karavasilis is funded by the Greek State Scholarships Foundation.

detected in every frame and associating detections between frames is converted into a linear programming problem which is solved using the k-shortest path algorithm. Partially occluded objects are also tracked in [22], where occlusion maps and prior knowledge on the objects' movement are used.

In this work, we present a tracking algorithm relying on the probabilistic representation of the object to be tracked and its subsequent localization in the image sequence. It is assumed that the appearance of the target may be described by a Gaussian mixture model (GMM) instead of a histogram or histogram signatures as it is the case, for instance, in [4], [6] and [9]. Moreover, the pixels of the target do not contribute equally to the likelihood of the target but they are weighted with respect to their distance from the center of the object. The localization of the target is obtained by maximizing the weighted likelihood along the frames of the image sequence. The numerical evaluation of the algorithm showed that it provides, in general, more accurate target localization than the Camshift algorithm [5] which is a variant of the mean shift algorithm [4] in OpenCV [23].

In the remaining of the paper, the tracking algorithm relying on the maximization of the weighted target likelihood is described in section II, experimental results are presented in section III and conclusions are drawn is section IV.

## II. Tracking by weighted likelihood

We assume that the object, which is represented by an ellipse, is known in the first frame of the image sequence. Using color and intensity features inside this ellipse a GMM is constructed by employing the EM algorithm.

Let $\boldsymbol{y}$ be a vector representing the coordinates of the center of the ellipse and $\boldsymbol{h}$ be a vector with components being the lengths of the major and minor axes of the ellipse. The coordinates of the $n$-th pixel is represented by $\boldsymbol{x}_n$ and the corresponding feature (i.e. color) by $\boldsymbol{I}_n$. We assign a weight $w_n(\boldsymbol{y})$ to every pixel by masking the ellipse with a kernel $k(\cdot)$:

$$w_n(\boldsymbol{y}) = k\left(f\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)\right), \tag{1}$$

where

$$f\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right) = \left(\frac{x_n^{(1)} - y^{(1)}}{h^{(1)}}\right)^2 + \left(\frac{x_n^{(2)} - y^{(2)}}{h^{(2)}}\right)^2 \tag{2}$$

where the superscripts 1 and 2 denote the horizontal and vertical coordinates respectively. Function $f$ eliminates the drawback of different axis length.

We now define the *weighted* log-likelihood function of the set of pixels inside an ellipse with center $\boldsymbol{y}$:

$$L(\boldsymbol{I},\boldsymbol{w}(\boldsymbol{y});\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{n=1}^{N} w_n(\boldsymbol{y}) L_n \tag{3}$$

where $N$ is the number of pixels inside the ellipse, $\boldsymbol{I} = \{\boldsymbol{I}_n\}_{n=1,\ldots,N}$, $\boldsymbol{w}(\boldsymbol{y}) = \{w_n(\boldsymbol{y})\}_{n=1,\ldots,N}$, where $w_n(\boldsymbol{y})$ denotes the importance of the $n$-th pixel to the model and the term

$$L_n = \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{I}_n;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \tag{4}$$

denotes the log-likelihood of a pixel described by a GMM of $K$ components with mixing proportions $\pi_k$, mean vectors $\boldsymbol{\mu}_k$ and covariance matrices $\boldsymbol{\Sigma}_k$, for $k = 1,\ldots,K$.

At first, the GMM parameters describing the target are computed by employing the EM algorithm for maximizing the weighted log-likelihood (3), leading to the following update formulas:

E-Step:

$$z_{k,n} = w_n(\boldsymbol{y}) \frac{\pi_k \mathcal{N}(\boldsymbol{I}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{I}_n|\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}. \tag{5}$$

M-Step:

$$N_k = \sum_{n=1}^{N} z_{k,n}, \tag{6}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} z_{k,n} \boldsymbol{I}_n, \tag{7}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} z_{k,n} \left(\boldsymbol{I}_n - \boldsymbol{\mu}_k\right)\left(\boldsymbol{I}_n - \boldsymbol{\mu}_k\right)^T, \tag{8}$$

$$\pi_k = N_k \frac{1}{\sum_{n=1}^{N} w_n}. \tag{9}$$

The above two steps are executed repeatedly until the log -likelihood (3) of two consecutive iterations do not change significantly or a maximum number of steps is reached. The initialization of the GMM parameters is done by using the K-means algorithm.

We consider that in the first frame the position of the target $\boldsymbol{y}$ and its size $\boldsymbol{h}$ are known and the tracking procedure consists in estimating $\boldsymbol{y}$ and $\boldsymbol{h}$ in the next frames. Computing the gradient of the weighted likelihood (3) with respect to $\boldsymbol{y}$ we obtain:

$$\frac{dL}{d\boldsymbol{y}} = \frac{dL(\boldsymbol{I},\boldsymbol{w}(\boldsymbol{y})|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}{d\boldsymbol{y}} = \sum_{n=1}^{N} \frac{df\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)}{d\boldsymbol{y}} L_n, \tag{10}$$

where

$$\frac{df\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)}{d\boldsymbol{y}} = \left[\frac{df\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)}{dy^{(1)}}, \frac{df\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)}{dy^{(2)}}\right]^T. \tag{11}$$

By defining the negative derivative of the kernel function as $g(x) = -\frac{dk(x)}{dx}$, we have:

$$\frac{df\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)}{d\boldsymbol{y}} = 2\boldsymbol{A}_n(\boldsymbol{y}) g\left(f\left(\boldsymbol{x}_n;\boldsymbol{y},\boldsymbol{h}\right)\right), \tag{12}$$

where

$$\boldsymbol{A}_n(\boldsymbol{y}) = \left[\frac{x_n^{(1)} - y^{(1)}}{h^{(1)^2}}, \frac{x_n^{(2)} - y^{(2)}}{h^{(2)^2}}\right]^T, \tag{13}$$

leading to:

$$\frac{dL}{d\boldsymbol{y}} = \sum_{n=1}^{N} 2\boldsymbol{A}_n(\boldsymbol{y}) g\left(f\left(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h}\right)\right) L_n. \qquad (14)$$

Once (14) is computed, we move along the gradient vector to one of its 8 neighboring pixels, as it is proposed in [6] in order to ensure a smooth motion between frames. An alternative would be to use the exact values of the gradient vector in case smooth motion is not a prerequisite. The advantage of using the weighted log-likelihood in (3) is that we obtain the gradient in (14) which depends on the target location $\boldsymbol{y}$. This is in contrast with a standard GMM-type likelihood (without the weight), which would not provide a gradient dependent on $\boldsymbol{y}$ and therefore the likelihood maximization with respect to it would not be feasible.

Due to the fact that the log-likelihood function (3) depends on the number of pixels $N$, we can not use it directly to evaluate the scale of the target. To overcome this drawback the number of pixels $N$, inside the ellipse, where the likelihood is evaluated must be constant. To this end, we only consider pixels in a certain grid. The distance of neighboring pixels in this grid may be increased or decreased in relation with the size of the ellipse. By these means, the number of pixels $N$ remains constant. This scale adaptation is performed independently in the horizontal and vertical directions and demands less computational resources compared to the computation of the position which necessitates the whole number of pixels inside the ellipse. The overall tracking procedure is described in the weighted likelihood tracking (WLT) algorithm 1.

**Algorithm 1** *WLT* algorithm

```
1:  function WLT(Image sequence)
2:      Input: an image sequence consisting of $T$ frames.
3:      Output: the ellipse center $\boldsymbol{y}$ at each frame.
4:      Initialization:
5:      Determine the initial position $\boldsymbol{y}_1$ and the size $\boldsymbol{h}_1$ of the target
6:      Compute the parameters $\pi_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of the GMM describing the target using (7), (8) and (9)
7:      Tracking:
8:      for frame $t = 2, \dots, T$ do
9:          $\boldsymbol{y}_t = \boldsymbol{y}_{t-1}$
10:         $\boldsymbol{h}_t = \boldsymbol{h}_{t-1}$
11:         while the likelihood in (3) increases do
12:             Move to $\boldsymbol{y}_t$ using (14)
13:         end while
14:         Estimate horizontal and vertical sizes of the target $\boldsymbol{h}_t = [h_t^{(1)}, h_t^{(2)}]^T$
15:     end for
16: end function
```

In this framework, the advantage of weighting the likelihood is twofold. Firstly, following the assumption adopted in kernel-based tracking methods [4], [6], [23], it considers that pixels near the center are more probable to belong to the object and they contribute more to the total likelihood. On the other hand, pixels which are more distant from the center may be part of the background and their contribution to the object's likelihood relatively smaller. Secondly, the main advantage of the weighted likelihood is that the weight at each pixel depends on the target location and the maximization of the likelihood is easily obtained with respect to it. This is not the case for a standard GMM likelihood function which cannot be employed in this framework.

## III. Experimental results

The evaluation of the proposed tracking algorithm was performed using six real datasets. The image sequences *Seq1*, *Seq2*, *Seq3* and *Seq4* are taken from the PETS'01 database while the datasets *Seq5* and *Seq6* are taken from PETS'06 database. In all of these image sequences the targets change their position and size simultaneously. The ground truth for these image sequences was manually determined (both for the size and the position of the target). Let us notice that although we show the ground truth delimited by rectangles, the WLT algorithm employs the inscribed ellipse.

As each object is represented by an ellipse, in order to evaluate the performance of a tracking algorithm we use the center and the size of the ellipse axis. We employ the evaluation criteria that were used in [6]. The first criterion is the number of frames in which the object is correctly tracked. An object is considered to be correctly tracked in a frame if the estimated rectangle covers at least 25% of the area of the target in the ground truth. The second criterion is the position error which is the Euclidian distance between the center of the object in the ground truth and the estimated target center, divided by the diagonal of the ground truth rectangle. Finally, the size error is defined as the Euclidian distance between the ground truth and the estimated vectors (with components being the width and the height of the ellipse), normalized by the ground truth length of the objects diagonal. The division with the diagonal of the object eliminates the problems due to different object sizes.

Let us notice that the choice of the kernel function $k(x)$ may theoretically affect the gradient of the log-likelihood (14). If we use a kernel with an exponential profile:

$$k(x) = \begin{cases} e^{(-x/\sigma)} & if \quad x \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad (15)$$

then the derivative of (14) becomes:

$$\frac{dL}{d\boldsymbol{y}} = \sum_{n=1}^{N} \boldsymbol{A}_n(\boldsymbol{y}) w_n(\boldsymbol{y}) L_n \qquad (16)$$

which involves $w_n(\boldsymbol{y})$. On the other hand, if we use an

Table I
THE PERFORMANCE OF THE WLT METHOD IN TERMS OF AVERAGE POSITION ERROR AND AVERAGE SIZE ERROR FOR AN EXPONENTIAL AND AN EPANECHNIKOV KERNEL. THE BOLD NUMBERS INDICATE THE BEST PERFORMANCE.

| Seq. | Frames Tracked | | Position Error | | Size Error | |
|---|---|---|---|---|---|---|
| | Exponential | Epanechnikov | Exponential | Epanechnikov | Exponential | Epanechnikov |
| *Seq1* | 499/499 | 499/499 | **0.102** | 0.193 | **0.230** | 0.357 |
| *Seq2* | 199/199 | 173/199 | **0.137** | 0.144 | 0.320 | **0.260** |
| *Seq3* | 299/299 | 299/299 | 0.182 | **0.116** | **0.211** | 0.239 |
| *Seq4* | 309/309 | 304/309 | **0.122** | 0.146 | **0.205** | 0.303 |
| *Seq5* | 129/129 | 98/129 | 0.130 | **0.097** | 0.337 | **0.201** |
| *Sqe6* | 169/169 | 169/169 | **0.149** | 0.167 | **0.274** | 0.335 |

Table II
THE PERFORMANCE OF THE WLT METHOD IN TERMS OF AVERAGE POSITION ERROR AND AVERAGE SIZE ERROR FOR DIFFERENT NUMBERS OF COMPONENTS $K$ EVALUATED FOR THE DATASET *Seq2*.

| $K$ | Frames Tracked | Position Error | Size Error |
|---|---|---|---|
| 2 | 199/199 | 0.135 | 0.273 |
| 3 | 199/199 | 0.137 | 0.320 |
| 4 | 199/199 | 0.136 | 0.292 |
| 5 | 199/199 | 0.134 | 0.302 |
| 6 | 199/199 | 0.116 | 0.254 |
| 7 | 199/199 | 0.137 | 0.318 |
| 8 | 199/199 | 0.129 | 0.304 |

Epanechnikov kernel:

$$k(x) = \begin{cases} \frac{1}{2}(1-x) & if \quad x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

then the derivative of (14) is simplified and becomes:

$$\frac{dL}{d\boldsymbol{y}} = \sum_{n=1}^{N} \boldsymbol{A}_n(\boldsymbol{y}) L_n. \tag{18}$$

In our experiments we used (15) and (16) with $\sigma = 1$ as the performance for this kernel type is slightly better (Table I).

We have also tested the proposed method for different numbers of components $K$ (Table II). For this particular example, the number of components $K$ does not seem to affect the performance of the algorithm. Even though the results presented in Table II correspond to *Seq2* dataset, the WLT method has similar performance for the rest of the datasets used in our experiments. The objects in these examples have few color components. For objects with more color components small values for $K$ may not be sufficient. On the other hand, the time needed by the tracking algorithm increases when the number of components increases. The representation of a target by a GMM is an efficient tool as a GMM may represent both the foreground and local background elements using different components.

We also compared our method with the OpenCV's implementation of Camshift algorithm [5], [23] which is a robust version of the mean shift algorithm [4] with scale adaptation. For Camshift, we used a 16 bin histogram for the hue component. We also did not take into account pixels with low or high brightness or low saturation (we apply thresholds equal to $10\%$ of the maximum pixel value) as it is suggested in [5]. For comparison purposes, we did not search for the rotation of the target in Camshift in order to have a common baseline. For the WLT algorithm we used $K = 3$ as it provides a relatively good tradeoff between the time needed and the flexibility of the GMM (Table II).

In Table III, the quantitative results of the compared methods are presented. The initial position in every sequence was manually determined in every algorithm. In *Seq1* and *Seq2*, where the targets are cars viewed from a fixed position camera, both algorithms successfully track the objects with Camshift having a slightly better performance. This results from the fact that the illumination conditions are the same for the whole image sequence. In *Seq3* and *Seq4*, the target is a car viewed from the rear from a moving camera under different illumination conditions. In *Seq3*, the color of the car is similar with the color of the road and Camshift did not estimate the position of the object accurately (the rectangle representing the target scaled up and included both the road and the car). Although we consider that Camshift tracked the target (the ground trough rectangle is inside the rectangle computed by Camshift), the position and size errors are large. In *Seq4*, Camshift fails to track the object after the half of the image sequence due to the fact that the color of the target is similar with the color of the background mountains. In contrast, WLT successfully tracks the objects in *Seq3* and *Seq4* despite these difficulties. The image sequences *Seq5* and *Seq6* are taken inside a subway using fixed position cameras with different viewpoint angles and show persons walking. In *Seq5*, a partial occlusion happens as another person walks between the camera and the target and in *Seq6* another person passes very close to the target. Both approaches successfully track the objects, with WLT showing a significantly better performance in terms of position and size errors.

In Fig. 1 and Fig. 2, qualitative result are presented. For every sequence, we demonstrate the ground trough which was used for evaluation (top row of each sequence) and the results obtained by using WLT (bottom row of each sequence). The left frame is the first frame of the sequence, while the other frames are uniformly sampled from the

Table III
THE PERFORMANCE OF THE COMPARED METHODS IN TERMS OF AVERAGE POSITION ERROR AND AVERAGE SIZE ERROR. THE BOLD NUMBERS INDICATE THE BEST PERFORMANCE.

| Seq. | Frames Tracked | | Position Error | | Size Error | |
|---|---|---|---|---|---|---|
| | Camshift | WLT | Camshift | WLT | Camshift | WLT |
| *Seq1* | 499/499 | 499/499 | **0.072** | 0.102 | **0.230** | **0.230** |
| *Seq2* | 199/199 | 199/199 | **0.080** | 0.137 | **0.234** | 0.320 |
| *Seq3* | 299/299 | 299/299 | 2.358 | **0.182** | 8.262 | **0.211** |
| *Seq4* | 165/309 | 309/309 | 3.000 | **0.122** | 3.316 | **0.205** |
| *Seq5* | 129/129 | 129/129 | 0.259 | **0.130** | 0.445 | **0.337** |
| *Seq6* | 169/169 | 169/169 | 0.262 | **0.149** | 0.421 | **0.274** |

sequence. As we can see, the results of WLT are close to the ground truth.

In these image sequences, the rectangles which represent the targets have dimensions around $150 \times 70$ pixels. For these target sizes, our algorithm, which is developed using OpenCV, runs in real time, as the average time needed for each frame is around $0.015$ sec (or equivalently 65 fps). The computer used during the experimental evaluation is a dual core pc (even though in the implementations we do not use the second core) at 1.83GHz with 2BG RAM at 667 MHz.

## IV. CONCLUSION

From the point of view of the target modeling and localization, the proposed algorithm belongs to the same family as the histogram based methods [4], [5], [7], [6], [23]. These methods minimize the distance between the probability distribution of the model and the distribution of the pixels at a candidate location in an image frame. The mean shift family of methods [4], [5] minimizes the Bhattacharyya distance while in [6], [7] the earth mover's distance is involved. The WLT method proposed herein, maximizes the weighted log-likelihood of the model without creating a second distribution in the image frame under consideration. The key issue in estimating the target's position is the weight term depending on the location of the target.

The method, in its current form, addresses the problem of single object tracking in real time. A perspective of this work is to integrate it into more sophisticated schemes including data association methods, for multiple object tracking and dynamic model inference schemes (e.g. update of the GMM parameters) to take into account changes in illumination conditions or partial occlusions.

## REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[2] E. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking," Freier Universitat Berlin, Institut fur Informatik, Tech. Rep. B 05-12, 2005.

[3] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[5] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, vol. Q2, 1998.

[6] Q. Zhao and H. Tao, "Differential earth mover's distance with its application to visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 274–287, 2010.

[7] V. Karavasilis, C. Nikou, and A. Likas, "Visual tracking using the earth mover's distance between Gaussian mixtures and Kalman filtering," *Image and Vision Computing*, vol. 29, no. 5, pp. 295–305, 2011.

[8] M. A. Carreira-Perpinan, "Gaussian mean shift is an EM algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 767–777, 2007.

[9] Z. Zivkovic and B. Krose, "An EM-like algorithm for color-histogram-based object tracking," in *2004 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2004, pp. 798–803.

[10] R. V. Babu, P. Pérez, and P. Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," *Image and Vision Computing*, vol. 25, no. 8, pp. 1205–1216, 2007.

[11] Z. Wang, X. Yang, Y. Xu, and S. Yu, "Camshift guided particle filter for visual tracking," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 407–413, 2009.

[12] M. Niethammer and A. Tannenbaum, "Dynamic geodesic snakes for visual tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004 (CVPR'04)*, vol. 1, 2004, pp. 660–667.

[13] E. Horbert, K. Rematas, and B. Leibe, "Level-set person segmentation and tracking with multi-region appearance models and top-down shape information," in *IEEE International Conference on Computer Vision (ICCV'11)*, 2011, pp. 1–8.

[14] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 266–280, 2000.

[15] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Particle filtering for geometric active contours with application to tracking moving and deforming objects," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005 (CVPR'05)*, vol. 2, 2005, pp. 2–9.

[16] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras, "Dependent multiple cue integration for robust tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 670–685, 2008.

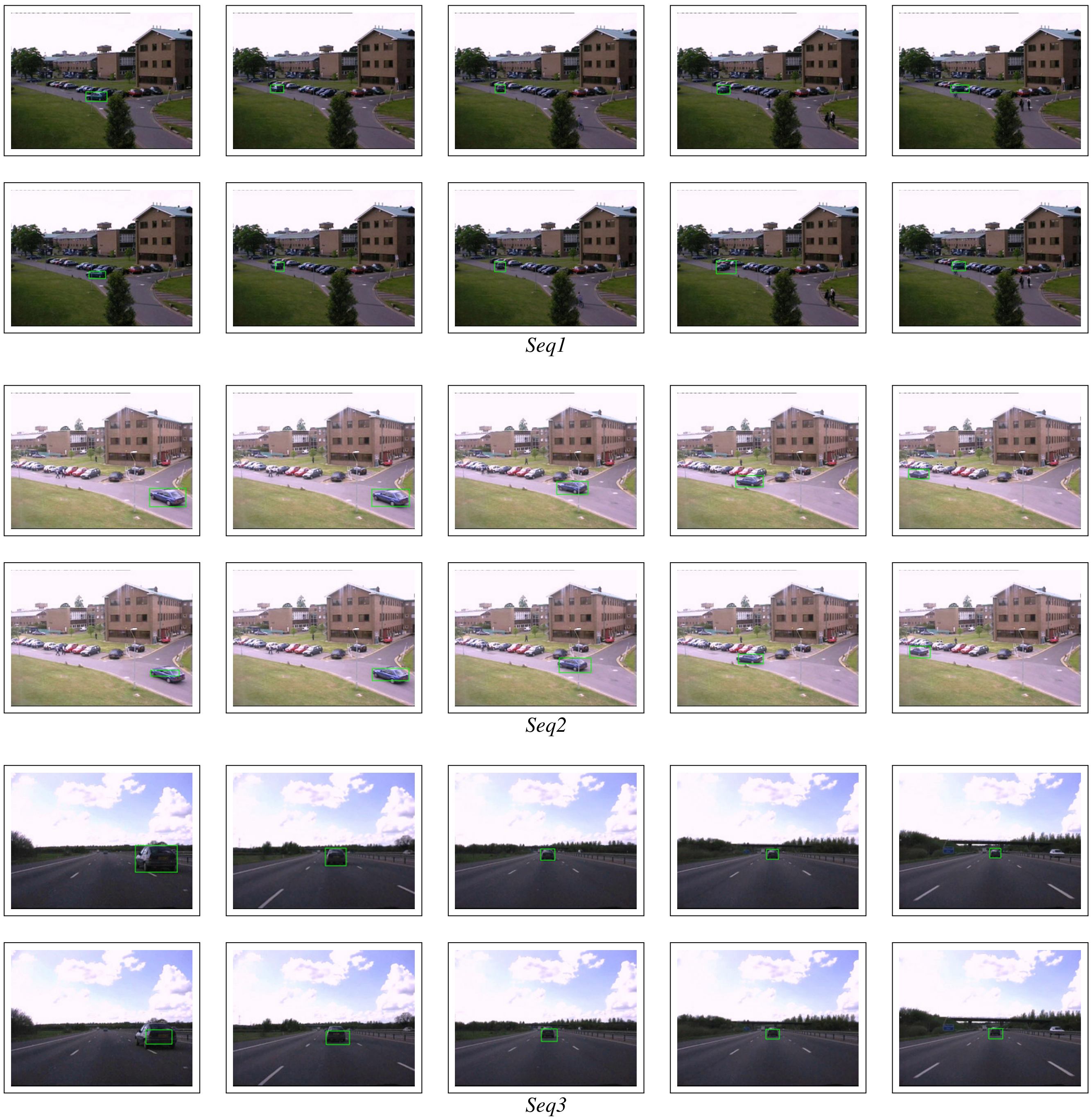

Figure 1. Representative results on the real datasets used in the experiments (*Seq1*, *Seq2* and *Seq3*). Although the inscribed ellipse is used in the computations, for visualization purposes, the target is bounded by a green rectangle. For every sequence, the top row shows the ground trough and the bottom rows show the results using WLT.

[17] J. Tu, H. Tao, and T. Huang, "Online updating appearance generative mixture model for mean-shift tracking," *Machine Vision and Applications*, vol. 20, no. 3, pp. 163–173, 2009.

[18] M. S. Allili and D. Ziou, "Object tracking in videos using adaptive mixture models and active contours," *Neurocomputing*, vol. 71, pp. 2001–2011, 2008.

[19] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[20] N. Papadakis and A. Bugeau, "Tracking with occlusions via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 144–157, 2011.

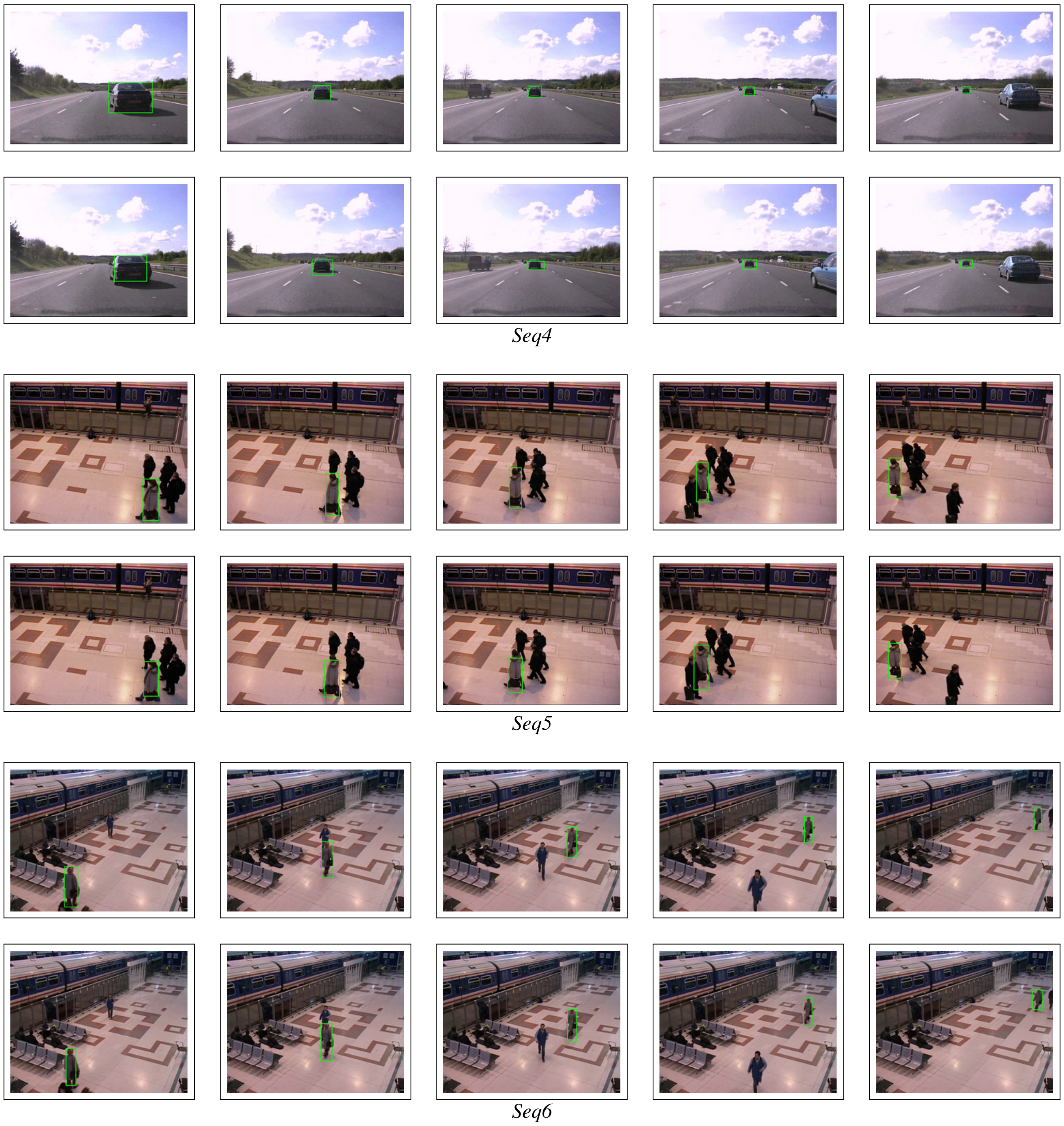

Figure 2. Representative results on the real datasets used in the experiments (*Seq4*, *Seq5* and *Seq6*). Although the inscribed ellipse is used in the computations, for visualization purposes, the target is bounded by a green rectangle. For every sequence, the top row shows the ground trough and the bottom rows show the results using WLT.

[21] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using "k"-shortest path optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1806–1819, 2011.

[22] V. Ablavsky and S. Sclaroff, "Layered graphical models for tracking partially occluded objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1758–1775, 2011.

[23] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reily, 2008.