2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing

Gaussian Mixture-based Mean Shift for Tracking under Abrupt Illumination Changes

Vasileios Karavasilis, Christophoros Nikou and Aristidis Likas
Department of Computer Science, University of Ioannina
PO Box 1186, 45110 Ioannina, Greece
Email: {vkaravas,cnikou,arly}@cs.uoi.gr

Abstract—**Mean shift is a fundamental algorithm for visual object tracking which is based on the minimization of the distance between the discrete histogram of the target and the discrete histogram of the neighborhood of a candidate image location. While the algorithm performs well when the target's appearance and the lighting conditions are constant, it may fail when these conditions are not met because the ideal histogram is generally shifted with respect to the reference histogram. In this work, we propose to compute the initial histogram of the target using a Gaussian mixture model (GMM) rather than impulses generated by simple counting. This mixture plays the role of a weighting function, in the histograms computed in subsequent frames, in order to make them smoother and increase the overlapping area with the initial histogram. By these means, sudden illumination changes between consecutive frames may exhibit smoother transitions between the two histograms and the involved distance is not trapped into local minima.**

I. Introduction

Visual tracking is the procedure of generating an inference about motion given a sequence of images. Based on a set of measurements in image frames the object's true position should be estimated. Tracking algorithms may be classified in two categories [1]. The first category is based on filtering and data association, while the second family of methods relies on target representation and localization.

The algorithms based on filtering assume that the moving object has an internal state which may be measured and, by combining the measurements with the model of state evolution, the object's position is estimated. A well known method of that category is the Kalman filter which successfully tracks objects even in the case of occlusion if the assumed type of motion is correctly modeled [2]. Another approach in this category are the particle filters [3] which include the Condensation [4] algorithm which is more general than Kalman filters, as it does not assume specific type of densities and, using factored sampling, it has the ability to predict an object's location under occlusion as well. Also, in this category, methods based on feature extraction and tracking were also proposed [5]. The object is represented by a set of scale invariant landmarks [6] which are tracked using optical flow [7]. The major drawback of these methods is that the type of object's movement should be correctly modeled.

On the other hand, tracking algorithms relying on target representation and localization employ a probabilistic model of the object appearance and try to detect this model in consecutive frames of the image sequence. More specifically, color or texture features of the object, masked by an isotropic kernel, are used to create a histogram. Then, the object's position is estimated by minimizing a cost function between the model's histogram and candidate histograms in the next image. A representative method in this category is the mean shift algorithm [1] where the object is supposed to be inside an ellipse and the histogram is constructed from pixel values inside that ellipse. This category also includes DEMD tracking algorithm [8] which represents the object by a histogram signature and uses the earth mover's distance (EMD) between histogram signatures in order to locate the object. A similar approach is presented in [9] where the object is represented by a Gaussian mixture model (GMM).

One drawback of these methods is that they can not handle total occlusions. Combinations of methods of the above two categories have been proposed in order to overcome their limitations. Mean shift combinations with Kalman filter [10] or particle filter [11] have been proposed.

Usually, tracking methods assume a simple shape for the object. 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]. Also, active contours have been integrated into other tracking methods such as particle filters [13] or into Bayesian filters to robustly segment the object from the background [14].

In this work, we propose a variant of the mean shift algorithm [1], in order to make the tracking procedure more robust to uniform or nonuniform abrupt light changes, such as flicker, light switch or shadow casts. In these cases, as the whole image becomes darker or brighter, the histogram of the target is shifted with respect to the initial histogram and the affinity between them would be close to zero, thus, making the mean shift algorithm to miss the object. In our approach, we propose to estimate the initial histogram of the target in the first frame by a Gaussian mixture model (GMM) and consider this mixture as a weighting function for the calculation of the histogram in the next frames. By these means, all of the mixture components contribute to the value of a specific bin and the histogram becomes smoother as its original values are diffused to neighboring bins.

In the remaining of the paper, the mean shift method is described in section II, the proposed evaluation of histogram

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

978-0-7695-4712-1/12 $26.00 © 2012 IEEE
DOI 10.1109/IIH-MSP.2012.129

IEEE computer society

is described in section III, experimental results are presented in section IV and the conclusions are drawn is section V.

II. Mean shift algorithm

The mean shift [1] is a target representation and localization algorithm trying to locate the object by finding the local maximum of a function. Here we give a brief review. The object target pdf is approximated by a histogram of m bins $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\ldots m}$, $\sum_{u=1}^{m} \hat{q}_u = 1$, with $\hat{q}_u$ being the u-th bin. To form the histogram, only the pixels inside an ellipse surrounding the object are taken into account. The center of the ellipse is assumed to be at the origin of the axes. Due to the fact that the ellipse contains both object pixels and background pixels, a kernel with profile $k(x)$, $k : [0, \infty) \to \Re$ is applied to every pixel to make pixels near the center of the ellipse to be considered more important. To reduce the influence of an eventual difference in the length of the ellipse axes, the pixel locations are normalized by dividing the pixel's coordinates with the ellipse's semi-axes lengths h_x and h_y. Let $\{\mathbf{x}_i^*\}_{i=1\ldots n}$ be the normalized pixel's spatial location. The u-th histogram bin is given by:

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|\mathbf{x}_i^*\|^2)\delta[b(\mathbf{x}_i^*) - u] \tag{1}$$

where $b : \Re^2 \to \{1 \ldots m\}$ associates each pixel with each bin in the quantized feature space, δ is the Kronecker delta function and C is a normalization factor such as $\sum_{u=1}^{m} \hat{q}_u = 1$.

In the next image, the object candidate is inside the same ellipse with its center at the normalized spatial location $\mathbf{y}$. Let $\{\mathbf{x}_i\}_{1\ldots n}$ be the normalized pixel coordinates inside the target candidate ellipse. The pdf of the target candidate is also approximated by an m-bin histogram $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}$, $\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1$, with each histogram bin given by

$$\hat{p}_u(\mathbf{y}) = D \sum_{i=1}^{n} k\left(\|\mathbf{y} - \mathbf{x}_i\|^2\right)\delta[b(\mathbf{x}_i) - u] \tag{2}$$

where D is a normalization factor such as $\sum_{u=1}^{m} \hat{p}_u(\mathbf{y}) = 1$.

The distance between $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}(\mathbf{y})$ is defined as:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \tag{3}$$

where

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} \tag{4}$$

is the similarity function between $\hat{\mathbf{q}}$ and $\hat{\mathbf{p}}(\mathbf{y})$ (Bhattacharyya coefficient).

To locate the object correctly in the image, the distance in (3) must be minimized, which is equivalent to maximize (4). The ellipse center is initialized at a location $\hat{\mathbf{y}}_0$ which is the ellipse center in the previous image frame. The probabilities $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$ are computed and using linear Taylor approximation of (4) around these values:

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{D}{2} \sum_{u=1}^{n} w_i k\left(\|\mathbf{y} - \mathbf{x}_i\|^2\right), \tag{5}$$

where

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}} \delta[b(\mathbf{x}_i) - u]. \tag{6}$$

As the first term of (5) is independent of $\mathbf{y}$, the second term of (5) must be maximized. The maximization of this term may be accomplished by employing the mean shift algorithm [1], which yields the following update:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n} \mathbf{x}_i w_i g\left(\|\hat{\mathbf{y}}_0 - \mathbf{x}_i\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\|\hat{\mathbf{y}}_0 - \mathbf{x}_i\|^2\right)}, \tag{7}$$

where $g(x) = -k'(x)$. The complete algorithm [1] is summarized in algorithm 1.

Algorithm 1 Mean shift tracking procedure

```
Input: The target model $\{\hat{q}_u\}_{u=1\ldots m}$ and its location $\hat{\mathbf{y}}_0$ in
       the previous frame.
1.     Initialize the center of the ellipse in the current frame
       at $\hat{\mathbf{y}}_0$, compute $\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1\ldots m}$ using (2).
2.     Compute the weights $\{w_i\}_{i=1\ldots n}$ according to (6).
3.     Compute the next location of the target candidate
       according to (7).
4.     If $\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\| < \epsilon$ Stop.
       Otherwise set $\hat{\mathbf{y}}_0 \leftarrow \hat{\mathbf{y}}_1$ and go to Step 2.
```

III. Target modeling by a GMM

If global, uniform or nonuniform, illumination changes take place, then the whole histogram $\hat{p}_u(\mathbf{y})$ in (2) will be (uniformly or not) shifted with respect to the initial histogram $\hat{q}_u$ in (1). For the sake of clarity, this issue is illustrated by a simple example in figure 1, where the initial histogram is shown in fig. 1(a) and the histogram of the target in the next frame (under abrupt illumination change) is shown in fig. 1(b). Notice that, ideally, this should be the histogram corresponding to the maximum of (3). However, by simple inspection, this distance is close to zero and the algorithm would respond with an erroneous image location for the target due to the influence of this distance in the computation of the weights in (6). Although this issue could be overcome for simple global uniform illumination changes (e.g. by subtracting the mean image value) the problem becomes more intricate if the involved changes in lighting conditions are highly non uniform. In figure 1(c), the GMM representing the density of the target in fig.1(a) is shown.

In order to estimate the GMM parameters we define the log-likelihood function of the color of pixels inside an ellipse:

$$L(\boldsymbol{I}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{I}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{8}$$

where $\boldsymbol{I} = \{\boldsymbol{I}_i\}_{i=1,\ldots,n}$ denote the color of every pixel, $\boldsymbol{\pi} = \{\pi_k\}_{k=1,\ldots,K}$ are the mixing proportions, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1,\ldots,K}$ are the mean vectors, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_k\}_{k=1,\ldots,K}$ are the covariance matrices and K denotes the number of the GMM components.

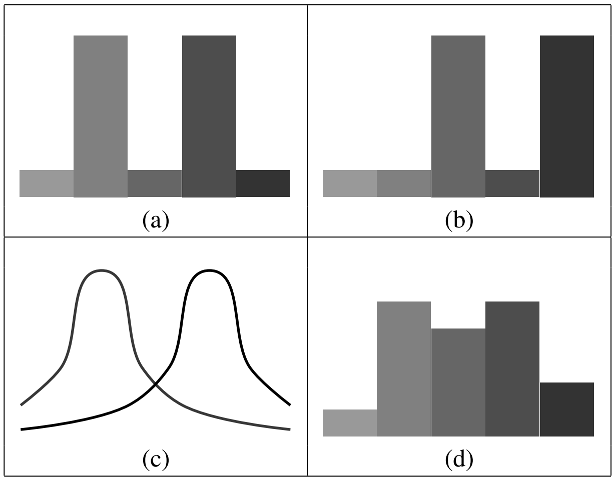


Fig. 1. a) The histogram of the target in the initial image. b) The histogram of the target in the next image is shifted due to an abrupt illumination change. c) The GMM of the target in the initial image. d) The resulting smooth histogram using (14).

The estimation of the GMM parameters is achieved through the EM algorithm [15]:

E-Step:

$$z_{k,i} = \frac{\pi_k \mathcal{N}(\boldsymbol{I}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{I}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{9}$$

M-Step:

$$\pi_k = \frac{\sum_{i=1}^{n} z_{k,i}}{n}, \tag{10}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{n} z_{k,i}\boldsymbol{I}_i, \tag{11}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{i=1}^{n} z_{k,i}\left(\boldsymbol{I}_i - \boldsymbol{\mu}_k\right)\left(\boldsymbol{I}_i - \boldsymbol{\mu}_k\right)^T, \tag{12}$$

The EM algorithm is employed in the first frame in order to estimate the GMM parameters. Having computed $\boldsymbol{\pi} = \{\pi_k\}_{k=1,\dots,K}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1,\dots,K}$ and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_k\}_{k=1,\dots,K}$ we can estimate the equivalent histograms' bins in (1) and (2).

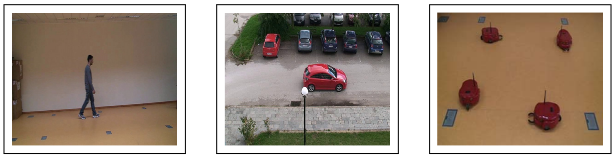
We assume that every bin is computed by the mixture of all components and the u-th histogram bin in (1) is now given by:

$$\hat{q}_u = C\sum_{i=1}^{n} k(\|\mathbf{x}_i^*\|^2)\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{I}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \tag{13}$$

Equivalently, in the next image the u-th histogram bin is given by:

$$\hat{p}_u(\mathbf{y}) = D\sum_{i=1}^{n} k\left(\|\mathbf{y}-\mathbf{x}_i\|^2\right)\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{I}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k), \tag{14}$$

which corresponds to the toy example in fig. 1(d). Notice that the transitions between bins are now smoother and the basin of attraction of the smoother histogram may be large enough to capture the reference histogram in fig. 1(a). The reference histogram looks now more similar to the histogram at the ideal target location.



Seq1 (300 frames) *Seq1* (700 frames) *Seq2* (250 frames)

Fig. 2. Representative frames of the datasets used in the experiments.

By following the same reasoning as in section II, we end up in the same update equation for y as in (7). However the weights w_i are given by:

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{y}}_0)}}\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{I}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \tag{15}$$

Therefore, the tracking procedure is the same as described in algorithm 1, but we use (13), (14) and (15) instead of (1), (2) and (6) respectively.

IV. Experimental results

The evaluation of the proposed tracking algorithm was performed using three datasets (Fig. 2). In *Seq1*, a man is walking from left to right, in *Seq2*, a car is moving from left to right and in *Seq3*, four robots are moving at different directions. The ground truth for these image sequences was manually determined. We compared our approach (referred as MSGMM) with the standard mean shift algorithm [1]. We use RGB images where and the number of the histogram bins is set to 16 in each channel, resulting to 16^3 bins totally.

We evaluated the performance of the tracking algorithm both in terms of position error and execution time. We define the position error as the average Euclidian distance between the ground truth's ellipse center and the ellipse estimated by the tracking algorithm (in normalized coordinates). The execution time is defined as the average time (seconds) needed per frame by the tracking procedure.

Firstly, we evaluate the proposed method for different numbers of the GMM components K (Table I). Comparing the position error for $K = 1,\dots,5$, we observe that the best results are obtained for $K = 2$. This happens due to the fact that the targets have relatively few colors. In terms of average time per frame, the fewer the components of the GMM are, the less execution time is needed. Here, we must point out that all these variants are executed in real time. For this set of experiments, we choose $K = 2$ for comparison with the standard mean shift in flicker conditions.

In order to evaluate the proposed method during illumination changes, we used six more sequences that are generated from sequences *Seq1*, *Seq2* and *Seq3*. The sequences with the subscript *a* are produced from the initial sequences by keeping the first frame the same and making the rest of the frames significantly brighter. The sequences with the subscript

TABLE I

The performance of the proposed method for different GMM components number (K) in terms of average position error and average execution time.

Sequece	$K=2$	$K=3$	$K=4$	$K=5$
	Position Error			
Seq1	0.216	0.226	0.242	0.270
Seq2	0.461	0.487	0.545	0.517
Sqe3	0.342	0.392	0.390	0.401
	Seconds/Frame			
Seq1	0.012	0.014	0.019	0.021
Seq2	0.013	0.017	0.019	0.024
Sqe3	0.010	0.012	0.014	0.016

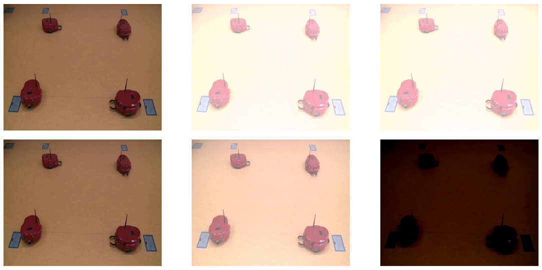


Fig. 3. Top row: the first frames of $\mathit{Seq3}_a$. Bottom row: the first frames of $\mathit{Seq3}_b$.

b are produced from the initial sequences by making the odd-numbered frames brighter and the even-numbered frames dimmer (Fig. 3).

In Table II, the comparative results between the standard mean shift algorithm and the proposed method with $K = 2$ GMM components are given. In normal conditions (*Seq1*, *Seq2* and *Seq3*), mean shift provides the best results in terms of position error in two out of three datasets. In terms of execution time, the MSGMM algorithm needs approximately twice the time needed by mean shift in all nine sequences. This results from the fact that computations involving the evaluation of the normal distribution for every pixel are performed.

When the lighting conditions change, MSGMM clearly outperforms mean shift. Especially, in $\mathit{Seq3}_a$ and $\mathit{Seq3}_b$, mean shift fails to track the object from the beginning. On the other hand, MSGMM successfully tracks the object with a position error of 0.919 for $\mathit{Seq3}_a$ and 1.029 for $\mathit{Seq3}_b$. The error around 1 means that the estimated center of the target is around the edge of the ellipse representing the object, but there is still common area between the ground truth ellipse and the estimated ellipse.

V. Conclusion

In this work, we modified the mean shift algorithm in order to make the tracking procedure more robust to illumination changes. We used Gaussian mixture model for the evaluation of histogram bins. This modification also affects the weights of every pixel during the tracking process. As shown by the experimental results, this approach can successfully track objects when the light conditions change dramatically. A future direction would be to employ this approach into other tracking algorithms involving histograms.

TABLE II

The performance of the compared methods (mean shift and MSGMM with $K = 2$) in terms of average position error and average size error.

	Position Error		Seconds/Frame	
Sequece	Mean shift	MSGMM	Mean shift	MSGMM
Seq1	0.264	0.216	0.007	0.012
$\mathit{Seq1}_a$	0.572	0.544	0.006	0.013
$\mathit{Seq1}_b$	0.904	0.875	0.012	0.026
Seq2	0.289	0.461	0.005	0.013
$\mathit{Seq2}_a$	0.608	0.251	0.006	0.018
$\mathit{Seq2}_b$	0.726	0.566	0.018	0.037
Sqe3	0.155	0.342	0.005	0.010
$\mathit{Seq3}_a$	-	0.919	0.005	0.015
$\mathit{Seq3}_b$	-	1.029	0.011	0.022

References

[1] 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.

[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. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1195–1209, 2009.

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

[5] C. J. Veenman, M. J. T. Reinders, and E.Backer, "Resolving motion correspondence for densely moving points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 54–72, 2001.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, 1981, pp. 674–679.

[8] 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.

[9] 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.

[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] 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.

[14] 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.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.