Visual Vehicle Tracking via Deep Learning and Particle Filter

Chapter · October 2020

DOI: 10.1007/978-981-15-6048-4_45

7 authors, including:

Hamd Ait Abdelali
Moroccan Foundation for Advanced Science, Innovation & Research
20 PUBLICATIONS **110** CITATIONS

Omar Bourja
Moroccan Foundation for Advanced Research Science and Innovation
19 PUBLICATIONS **93** CITATIONS

Rajae Haouari
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
11 PUBLICATIONS **19** CITATIONS

Hatim Derrouz
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
15 PUBLICATIONS **90** CITATIONS

Visual Vehicle Tracking via Deep Learning and Particle Filter

Hamd Ait Abdelali, Omar Bourja, Rajae Haouari, Hatim Derrouz, Yahya Zennayi, François Bourzex, and Rachid Oulad Haj Thami

Abstract Visual vehicle tracking is one of the most challenging research topics in computer vision. In this paper, we propose a novel and efficient approach based on the particle filter technique and deep learning for multiple vehicle tracking, where the main focus is to associate vehicles efficiently for online and real-time applications. Experimental results illustrate the effectiveness of the system we are proposing.

Keywords Computer vision · Vehicles tracking · Computer vision · Deep learning · YOLO · Bhattacharyya kernel · Histogram-based · Particle filter · IoU metric

1 Introduction

Visual vehicle tracking has received much attention in the past decade since it becomes a vital component in intelligent transportation surveillance systems. Visual vehicle tracking approaches aim to associate the target vehicles in consecutive frames of video. The core issue of those approaches is their dependency on the target vehicle behavior: In the case where the vehicle's speed is relatively higher, then the frame rate or the orientation changing during the tracking process can have a severe impact on the model's performance. A typical visual vehicle tracking system consists of five

H. Ait Abdelali (✉) · O. Bourja · R. Haouari · H. Derrouz · Y. Zennayi · F. Bourzex
Embedded System and IA Department, MAScIR, 10100 Rabat, Morocco
e-mail: h.aitabdelali@mascir.com

O. Bourja
RIME Laboratory, Mohammadia School of Engineers, Mohammed V University, 10000 Rabat, Morocco

R. Haouari · H. Derrouz · R. Oulad Haj Thami
ADMIR Laboratory, Rabat IT Center, IRDA Team, ENSIAS, Mohammed V University, 10000 Rabat, Morocco

Y. Zennayi
Smart Communications Research Team, Department of Electrical Engineering at Mohammadia School of Engineers, Mohamed V University, 10000 Rabat, Morocco

F. Saeed et al. (eds.), *Advances on Smart and Soft Computing*, Advances in Intelligent Systems and Computing 1188, https://doi.org/10.1007/978-981-15-6048-4_45

essential components: object representation, dynamic model, object location, search mechanism, and data association. Based on the implementation of those components, visual vehicle tracking approaches can be categorized in different manners. The object representation component is responsible for the process of matching the target appearance under different influencing factors and determine the objective function to be used for finding the object in the frames. A dynamic model is used to predict the possible object. Since it is difficult to adapt an efficient dynamic model for fast movements and due to faster processors, the most current tracking algorithms use a random walk model to predict likely states. The main ideas of this work are: We leverage the power of the deep learning approach You Only Look Once (YOLO) [1]-based detection in the context of multiple vehicle tracking, a pragmatic tracking approach based on the particle filter [2–5], and the Bhattacharyya [6–8] kernel is presented and evaluated.

The rest of the work is organized as follows: Sect. 2 describes related and existing work in this field. Sections 3.2 and 3.3 presents basic particle filter algorithm and Bhattacharyya kernel method. Section 4 introduces the proposed approach. The experimental results are presented in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

The Visual object tracking problem is formulated as a sequential recursive estimation of the probability distribution of the object in the previous frame. In the literature, researchers have proposed a variety of tracking model-based. In recent years, approaches based on the particle filter algorithm start gaining increasing interest, especially for real-time applications, such as traffic monitoring. This interest could be explained by the fact that the particle filter algorithm has shown a significant capability of nonlinear, non-Gaussian, and multi-model data processing. In [9], the authors presented the integration of color distributions in particle filter. In this work, for the measure of similarity between two distributions of color, the Bhattacharyya coefficient is applied. This approach has the ability to work in real time, but the processing time depends on the size of the region and the number of samples. The authors of [10] presented a color-based method for object tracking and developed an efficient tracking algorithm based on particle filter. To accelerate the weight computation of each particle, the integral images are used to compute the histogram. The main weakness of this approach is the large storage capacity. Similarly, Li et al. [11] also used a particle filter in combination with an adaptive fusion method for visual object tracking. In this work, the proposed method is more robust to illumination changes, particle occlusions, pose variations, cluttered backgrounds, and camera motion. However, the used dataset is limited comparing to other works. The research in [12] describes a novel approach that tackles the particle filter sample impoverishment problem. The main limitation of this approach is its dependency on the object speed since its performance decreases in the case where the speed of the moving object is higher than the frame rate. In [13], the authors used a hybrid estimation algorithm based on the particle filter method. This algorithm used the standard color of the particle filter in two steps: In the first step, the proposed algorithm handled the non-rigid and deformation of object, and in the second, it clutters the background. The major defect of this algorithm is that the number of particles increases with the increase of the number of tracked objects, which impacts the capability of the algorithm in real time, since the more significant the size of the state vector, the higher the competition time needed. The authors in [14] suggested the use of an immune genetic algorithm (IGA) to optimize the particle set. This optimization aims to describe the state of the target more significantly. Moreover, the use of this optimization technique increases the number of important particles, improves particle diversity, and decreases the error of state estimation and video tracking. The authors in [2] also proposed an optimized algorithm based on a continuously adaptive mean-shift (CamShift) method. The main idea of this work is improving the tracking accuracy by exploring the interaction between particle filtering and CamShift outputs. The use of the CamShift method helps improve the performance of the particle filter in sampling the position and scale space and reduce computing and execution time. Another optimization approach is presented in [15]. In this work, the authors used a particle filter with a posterior mode tracker to track target objects in the presence of illumination variations in the video frames. Similarly, in [16], the authors developed an object tacking approach that handled not only the illumination variations, but also the pose changes and the partial occlusions, as well. In [3], a Feature-driven (FD) motion-based model for object tracking is proposed. The proposed model uses features from the accelerated segment test (FAST) and matches them between the video sequences frames. This model was mainly developed to handle the abrupt motion-tracking problems that considered one of the significant challenges of tracking approaches. The main downside of this model is that it only uses the location cues for guiding particle propagation.

Recently, researchers start using deep learning techniques [17–19], to extract competent deep characteristics. In many complicated tasks relied on handcrafted features such as monitoring, location, traffic crowd detection, identification, self-stabilization, crash avoidance, obstacle, and vehicle tracking, deep learning algorithms have exhibited impressive results. With the smart video surveillance, autonomous vehicles, and numerous people-counting applications, facial recognition, the demand for rapid and accurate object detection systems is becoming a necessity. Such systems require not only the classification and identification but also the localization of each object in the image, which makes objects identification a much harder task than their conventional counterparts in computer vision.

3 Deep Learning

3.1 YOLO Detector

YOLO stands for You Only Look Once [1]. This technique watches at the image just for once and processes it simultaneously. Rather than of a large softmax, YOLO uses several softmax as a hierarchical tree; each softmax determines for a similar group of objects. YOLO is considered to be one of the fastest and most used object detection algorithms to predict several bounding boxes and their category probabilities at the same time YOLO use a single convolutional network.

3.2 Particle Filter

Particle filter is a sequential Monte Carlo for estimating the posterior distribution. It includes two parts: observation model and system model. See Fig. 1:

To predict the possible position of the targets, we use a system model given by:

$$X_t = g(X_{t-1}) + U_t \tag{1}$$

where $g(.)$ is the transition function, X_t presents the state of targets at time t, and U_t is the system noise. The observation model is used to determine the locations of the targets and has the form:

$$Y_t = f(X_t) + V_t \tag{2}$$

where $f(.)$ is a measurement function and V_t system noise. Let Y_t and X_t measurements and the hidden state of the object of interest at discrete time t, respectively. For tracking, the filtering distribution of interest is presented by $p(X_t|Y_{1:t})$, where $Y_{1:t} = (Y_1, \ldots, Y_t)$ is the observations in the current time. The estimation of filtering distribution content two-step recursion:

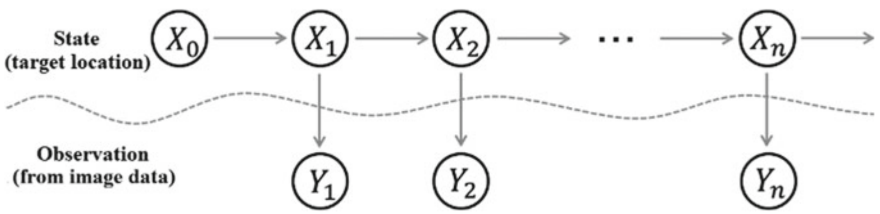

Fig. 1 Observation model and system model

$$p(X_t|Y_{1:t-1}) = \int p(X_t|X_{1:t-1})p(X_{t-1}|Y_{1:t-1})\mathrm{d}X_{t-1} \tag{3}$$

$$p(X_t|Y_{1:t}) = p(Y_t|X_{1:t})p(X_t|Y_{1:t-1}) \tag{4}$$

where $p(X_t|Y_{1:t-1})$ presented the state evolution and $p(Y_t|X_{1:t})$ is the current observation. To this end, let given a weighted set of samples $\left\{X_{t-1}^{(i)}, W_{t-1}^{(i)}\right\}_{i=t}^{N}$ approximately according to $p(X_{t-1}|Y_{1:t-1})$, generated form a proposal distribution, $X_t^{(i)} \sim q(X_t|X_{t-1}, Y_t)$, $i = 1, \ldots, N$, the new weights are set to:

$$W_t^{(i)} = W_{t-1}^{(i)} \frac{p(Y_t|X_t^{(i)})p(X_t^{(i)}|X_{t-1}^{(i)})}{q(X_t|X_{t-1}^{(i)}, Y_t)}, \quad \sum_{i=1}^{N} W_t^{(i)} = 1. \tag{5}$$

The new particle set $\left\{X_t^{(i)}, W_t^{(i)}\right\}_{i=t}^{N}$ is then approximately distributed according to $p(X_t|Y_{1:t})$.

3.3 Bhattacharyya Kernels

The Bhattacharyya kernel is a perfect similarity measure for discrete distributions. Let h_V be the histogram of vehicle tracking, and the number of pixels inside the targets V as $|V|$, which is also equal to $|V| = \sum_k h_V(k)$ the sum over bins. Let q be the normalized histogram obtained in the frames and p be the normalized version of the targets h_V defend by $p = \frac{h_V}{|V|}$, and $\delta[n]$ is the Kronecker delta:

$$h_V(k) = \sum_{x \in V} \delta[b(x) - k], \tag{6}$$

where $\delta[n] = 1$ if $n = 0$, and $\delta[n] = 0$ otherwise. The Bhattacharyya kernel can be expressed as:

$$\begin{aligned}
\mathrm{Bhat}(p, q) &= \sum_k \sqrt{p(k)q(k)} \\
&= \sum_k \sqrt{p(k)\left(\frac{1}{|V|}\sum_{x \in V}\delta[b(x) - k]\right)} \\
&= \frac{1}{\sqrt{|V|}} \sum_{x \in V} \sum_k \sqrt{p(k)\delta[b(x) - k]} \\
&= \frac{1}{\sqrt{|V|}} \sum_{x \in V} \sqrt{p(b(x))}.
\end{aligned} \tag{7}$$

where $b(x)$ is a map function a pixel x to its corresponding bin index, the computation of the Bhattacharyya kernel taking the sum of values of $\sqrt{p(b(x))}$ within the target V.

4 Proposed Approach

The proposed system is a filtering method based on recursive Bayesian estimation. The core idea of this work is that the density distribution is presented using random sampling of particles, with a nonlinear problem. The working mechanism of the proposed system has four major steps, namely selection step, prediction step, likelihood step, and measurement step. These steps are shown in Fig. 2:

Firstly, we generate N particles in an area that contains the possible locations of the targets. This step named selection or initialization. During the prediction step, each particle is transformed according to the state model by insertion random noise to emulate the impact of noise on the state. In the likelihood step, the weight of apiece particle is calculated, and the similarity (using Bhattacharyya kernel) between the target model and candidate regions is described. Furthermore, in this step, the weights are normalized to ensure that the sum of the weights is equal to 1. Finally, in the measurement step, the weight of each particle is re-examined based on the new sate. We assess the likelihood probability and re-sampling the particles for the next iteration. The steps of the proposed approach are presented in Fig. 3:

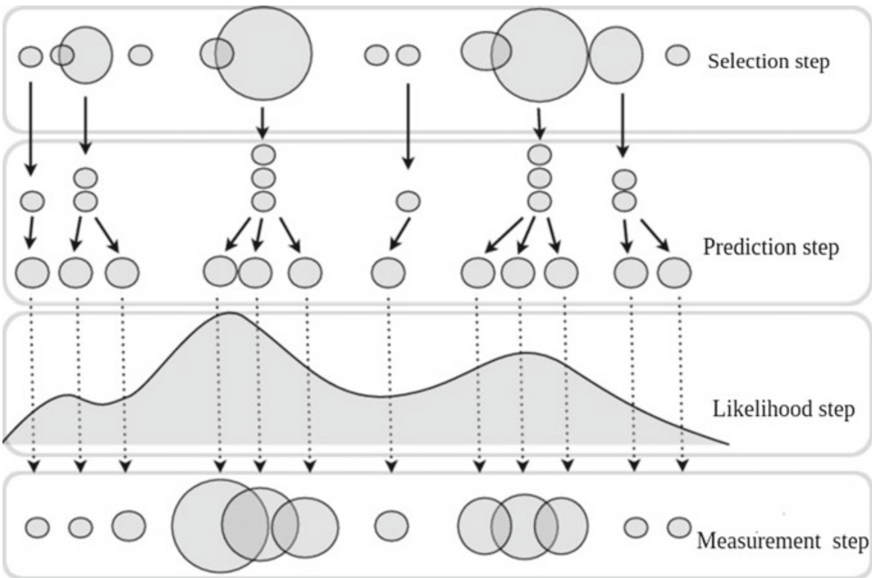

Fig. 2 Methodology of particle filter for vehicles tracking

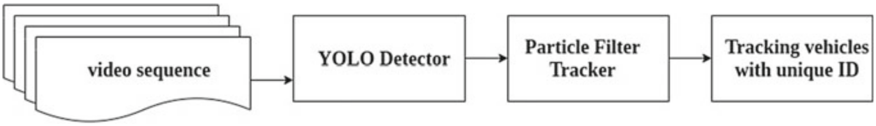

Fig. 3 Flowchart representation for visual vehicles detection and tracking

The Algorithm of the proposed method can be explained as follows:

Step 1. For each frame, the regions of targets (vehicles) are detected by the YOLO detector, and the sample set of N particles.

Step 2. Prediction the localisation's of the targets using second-order auto-regressive dynamics, for evaluate how the state of an targets will change by feeding it through a observation and system model, using Eqs. (1) and (2).

Step 3. Compute the Likelihood of each particle, and calculate the similarity between targets model and candidates regions in the current frame, using Eqs. (5) and (7).

Step 4. Measurement the weight of each particle is re-examine created on the new frame, we estimate the Likelihood, and Resampling $\left\{X_n^{(i)}, W_n^{(i)}\right\}_{i=n}^{N}$, and then approximately distributed according to $p(X_n|Y_{1:n})$.

Step 5. Draw trajectory of the targets by ID in the current frame, and go to step 2.

5 Experiment Result

This section shows the tracking results using the proposed method. The proposed approach components are run at 3.70 GHz single core of an Intel(R) Core (TM) i7-8700 K machine with 16 GB memory, NVIDIA GeForce GTX 1080 Ti, and Operating System Linux 64-bit. The program was implemented using C++, Qt Framework, and the Open Source Computer Vision (OpenCV) library. To check the performance effectiveness of the proposed approach, we use Intersection over Union (IoU) metric [20], also known as the Jaccard index, to compare the similarity between two arbitrary shapes (the predicted bounding box and the ground truth). Intersection over Union encodes the shape properties of the objects under comparison, e.g., the widths, heights, and locations of two bounding boxes. In the experiment, we use three videos; each has its characteristics.

We first use a road video (where the frame rate is 15 fps, the resolution is 2456 × 2054, and the number of frames is 1225) to check the effectiveness of the proposed approach. As shown in Fig. 4, the external bounding box red color represents the real

Fig. 4 Tracking results of the road sequence. Frames 1 and 78 are displayed. The predicted bounding box is drawn in blue, while the ground-truth bounding box is drawn in red. The metric IoU is higher than 0.95

trajectory of the vehicle (the ground truth, i.e., hand labeled), and the blue bounding box represents the trajectory of the vehicle by the proposed approach.

The second test is an Intersection_1 video (where the frame rate is 15 fps, the resolution is 2456×2054, and the number of frames is 1106), used to check the effectiveness of the proposed approach on a more complicated situation Fig. 5. The vehicles exhibit large-scale changes with partial occlusion. However, the proposed approach performs better in estimating the orientation and the scale of the targets, especially when an occlusion occurs.

The last experiment is an Intersection_2 video (where the frame rate is 15 fps, the resolution is 2456×2054, and the number of frames is 1124) presented in Fig. 6. The results show that the proposed method estimates the good accuracy of orientations and the scales of the targets.

The results show that the proposed approach is robust to detect and track the trajectory of the vehicles in different situations (pause, scale variation, occlusion, and rotation).

Fig. 5 Tracking results of the Intersection$_1$ sequence. Frames 1 and 172 are displayed. The predicted bounding box is drawn in blue, while the ground-truth bounding box is drawn in red. The metric IoU is higher than 0.80

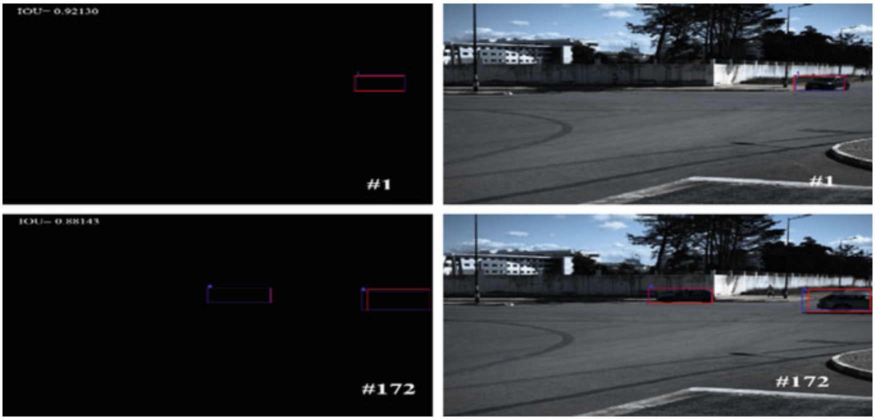

Fig. 6 Tracking results of the Intersection$_1$ sequence. Frames 69, and 112 are displayed. The predicted bounding box is drawn in blue, while the ground-truth bounding box is drawn in red. The metric IoU is higher than 0.88

6 Conclusion

In this paper, a novel visual vehicle tracking algorithm has been presented. The detection of moving vehicles is done using the YOLO detector and the particle filter algorithm for tracking vehicles in consecutive frames of a video sequence. In this method, we combine the Bhattacharyya kernels and the particle filter. Numerous experiments include challenging video when the proposed algorithm achieves excellent results.

References

1. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer, Cham (2016)
2. Ma, J., Han, C., Chen, Y.: Efficient visual tracking using particle filter. Inform. Fusion (2007)
3. Nummiaro, K., Koller-Meier, E., Van Gool, L.: Object tracking with an adaptive color-based particle filter. Pattern Recogn., 353–360 (2002)
4. Abdalmoaty, M.: Identification of stochastic nonlinear dynamical models using estimating functions. Dissertation. KTH Royal Institute of Technology (2019)
5. Li, A., Jing, Z., Hu, S.: Particle filter based visual tracking with multi-cue adaptive fusion. Chin. Opt. Lett. **3**(6), 326–329 (2005)
6. Tony, J., Risi, K.: Bhattacharyya expected likelihood kernels. Paper Presented at the Meeting of the COLT (2003)
7. Tony, J., Risi, K., Andrew, H.: Probability product kernels. J. Mach. Learn. Res. **5**, 819–844 (2004)
8. Chang, H.-W., Chen, H.-T.: A square-root sampling approach to fast histogram-based search. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3043–3049, 13–18 June 2010
9. Vadakkepat, P., Jing, L.: Improved particle filter in sensor fusion for tracking randomly moving object. IEEE Trans. Instrum. Meas. **55**(5), 1823–1832 (2006)
10. Czyz, J., Ristic, B., Macq, B.: A particle filter for joint detection and tracking of colour objects. Image Vis. Comput. **25**, 1271–1281 (2007)
11. Han, H., Ding, Y., Hao, K., Liang, X.: An evolutionary particle filter with the immune genetic algorithm for intelligent videotarget tracking. Comput. Math. Appl. **62**(7), 2685–2695 (2011)
12. Samarjit Das, N.V., Kale, A.: Particle filter with a mode tracker for visual tracking across illumination changes. IEEE Trans. Image Process. **21**(4), 2340–2346 (2012)
13. Ho, M., Chiang, C., Su, Y.: Object tracking by exploiting adaptive region-wise linear subspace representations and adaptive templates in an iterative particle filter. Pattern Recogn. Lett. **33**(5), 500–512 (2012)
14. Wang, Z., Yang, X., Xu, Y., Yu, S.: CamShift guided particlefilter for visual tracking. Pattern Recogn. Lett. **30**(4), 407–413 (2009)
15. Wang, W., Wang, B.: Feature-driven motion model-based particle-filter tracking method with abrupt motion handling. Opt. Eng. **51**, 4 (2012)
16. Mallikarjuna Rao, G., Satyanarayana, C.: Visual object target tracking using particle filter: a survey. IJIGSP **5**(6), 57–71 (2013)
17. Moon, S., Lee, J., Nam, D., Kim, H., Kim, W.: A comparative study on multi-object tracking methods for sports events. In: 19th International Conference on Advanced Communication Technology (ICACT) (2017)
18. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and real time tracking. In: IEEE International Conference on Image Processing, Phoenix, AZ, pp. 3464–3468 (2016)
19. Mekonnen, A., Lerasle, F.: Comparative evaluations of selected tracking by detection approaches. IEEE Trans. Circuits Syst. Video Technol. **29**(4), 996–1010 (2019)
20. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)