

# A Distributed Particle Filter Applied in Single Object Tracking

Di Wang, Min Chen

Department of Computer Science, State University of New York at New Paltz, New York, USA

Email: di.wang4@ibm.com, chenm@newpaltz.edu

**How to cite this paper:** Wang, D. and Chen, M. (2024) A Distributed Particle Filter Applied in Single Object Tracking. *Journal of Computer and Communications*, 12, 99-109.

<https://doi.org/10.4236/jcc.2024.128006>

**Received:** July 18, 2024

**Accepted:** August 19, 2024

**Published:** August 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Visual object-tracking is a fundamental task applied in many applications of computer vision. Particle filter is one of the techniques which has been widely used in object tracking. Due to the virtue of extendability and flexibility on both linear and non-linear environments, various particle filter-based trackers have been proposed in the literature. However, the conventional approach cannot handle very large videos efficiently in the current data intensive information age. In this work, a parallelized particle filter is provided in a distributed framework provided by the Hadoop/Map-Reduce infrastructure to tackle object-tracking tasks. The experiments indicate that the proposed algorithm has a better convergence and accuracy as compared to the traditional particle filter. The computational power and the scalability of the proposed particle filter in single object tracking have been enhanced as well.

## Keywords

Distributed System, Particle Filter, Single Object Tracking

## 1. Introduction

Visual object-tracking is a very active research topic within the area of computer vision. One of the reasons is the increasing need for the automated analysis of videos. Object-tracking is applied to many areas [1] including motion-based recognition, automatic surveillance, video indexing, human-computer interaction, traffic monitoring, vehicle navigation, etc.

Object tracking is a challenging task in many computer vision applications including surveillance, gesture recognition, vehicle tracking, medical imaging, etc [1]. Object tracking aims to identify the target state using trajectory, orientation, position and scale within a sequence of video. In particular, the tracking agent estimates the moving object frame by frame and labels the area of interest in the

video frame based on the previous state [2]. Object tracking can be classified into two groups from the application point of view: general object tracking and tracking application. The general object tracking aims to explore the methods of tracking different types of objects using a unified framework to simulate the human vision and cognition. The tracking application aims to explore the methods for specific application scenarios, such as pedestrian detection, face detection, text detection, etc. New techniques in deep learning lead to remarkable breakthroughs into object tracking. The challenges of object tracking arise due to the noise in images, the complexity of object movements, the illumination of changing scenes and the difficulty of defining object shapes. In order to simplify the task, most approaches make the assumption that no abrupt changes between the object movements. In addition, the object is moving in a constant velocity with an acceleration based on the previous knowledge.

In light of the differences of object representation, image feature, model of motion, appearance and shape of objects [3], a variety of tracking algorithms have been proposed in object tracking. Generally, there are two categories: deterministic approaches and stochastic approaches. Deterministic approaches typically use an iterative search to track the local optima with a cost function. The cost function measures the similarity between the current image and the template image. On the other hand, stochastic approaches model the underlying dynamics of the tracking system using state space. There is exactly one mode in the posterior probability density function (pdf) in a linear-Gaussian model with linear measurement [4]. Kalman filter is one popular method to propagate and update the mean and covariance of the distribution. However, it is impossible to estimate the distribution analytically in nonlinear or non-Gaussian problem. Sequential Monte Carlo, also known as particle filter, is a widely used approach which recursively constructs the pdf of the state space using Monte Carlo integration [5].

Particle filter is easy to extend and it provides the flexibility to handle both non-linear and non-normal object models. Contours, color features and appearance models are commonly used in the particle filter approach [6]. Particle filter approximates the filtered pdf by a set of weighted particles in any state model. The likelihood score is used to measure the weighted particles. Tremendous amount of videos from all applications have been accumulated, and how to analyze these videos efficiently and on time is challenging. However, most of the particle filter-based algorithms have paid little attention on how to parallelize the framework in order to handle the large volume of videos in a reasonable time. Recently, MapReduce framework has gained significant momentum from industry and academic because of its simplicity, scalability, and fault tolerance. This programming paradigm is a scalable and fault-tolerant data processing tool that was developed to provide significant improvements in large-scale data-intensive applications in clusters. Based on the aforementioned reasons, a parallel particle filter is conducted in a distributed framework provided by the

Hadoop/Map-Reduce infrastructure in this work. We aim to propose a particle filter with Hadoop/MapReduce to enhance the computational power and the scalability of a parallel particle filter in single-object tracking. In addition, this paper is an extended version of [7].

The rest of the paper is organized as follows: related work is discussed in 2. The proposed MapReduce-based algorithm is demonstrated in Section 3. The experimental results and analysis are described in Section 4, and the paper is concluded in Section 5.

## 2. Related Work

Different detectors have been presented in the literature. In [8], a real-time detection of human faces is proposed for the first time without any constraints. The proposed Viola-Jones detector, which was running on a Pentium III CPU, was tens or even hundreds of times faster than any other algorithms in its time. It used sliding windows to go through all possible locations and scales in an image to see if any window contains a human face. However, the calculation behind it was far beyond the computer's power of its time. Histogram of Oriented Gradients (HOG) feature descriptor was originally proposed by [9] in 2005. The HOG descriptor was computed on a dense grid of uniformly spaced cells and used overlapping local contrast normalization to improve accuracy. The HOG was motivated primarily by the problem of pedestrian detection. To detect objects of different sizes, the HOG detector rescales the input image for multiple times while keeping the size of a detection window unchanged. Deformable Part-based Model (DPM) is an extension of the HOG detector which was proposed by [10] in 2008. The DPM consists of a root-filter and a number of part-filters with a weakly supervised learning method.

Object detection has reached a plateau after 2010 due to the performance of hand-crafted features becoming saturated. The rebirth of convolutional neural networks is able to learn robust and high-level feature representations of an image. Object detection can be classified into two groups: two-stage detection and one-stage detection.

[11] proposed a two-stage detector named RCNN. It used a selective search to extract a set of object proposals. Each proposal is rescaled to a fixed size image and fed into a CNN model to extract features. A linear SVM classifier is used to predict the presence of an object within each region in order to recognize object categories. However, the drawbacks are obvious for RCNN since the redundant feature computations on a large number of overlapped proposals lead to an extremely slow detection speed. In 2014, [12] proposed a Spatial Pyramid Pooling network (SPPNet) to overcome this problem. It introduced a Spatial Pyramid Pooling (SPP) layer to enable a CNN to generate a fixed-length representation regardless of the size of image/region of interest without rescaling it. However, the training for SPPNet is still multi-stage and it only fine-tunes its fully connected layers while ignoring all previous layers. [13] proposed a Fast RCNN de-

detector to further improve RCNN and SPPNet. The Fast RCNN simultaneously train a detector with a bounding box regressor under the same network configurations. It has successfully integrates the advantages of RCNN and SPPNet. However, the speed is still a challenging problem for the proposed algorithm.

The first one-stage detector was proposed by [14] in 2015. The proposed detector YOLO used a single neural network to the full image. It has reached a great improvement of detection speed. It divides the image into regions and predicts bounding boxes and probabilities for each region simultaneously. However, compared with two-stage detectors, YOLO suffers from a drop of the localization accuracy, especially for some small objects. [15] introduced the multi-reference and multi-resolution detection techniques to significantly improve the detection accuracy of a one-stage detector, especially for some small objects. Despite of its high speed and simplicity, the one-stage detectors were under-performed two-stage detectors for years. The reason is because the extreme foreground-background class imbalance encountered during training of dense detectors [16].

The idea of a particle filter was independently proposed by different groups. The particle filter based tracking techniques usually use contours, color features, or appearance models [17]. However, the majority of these techniques ignores the spatial layout information. In addition, the computation cost is expensive if the tracked region and the number of samples are large. The parallel techniques have been used in face detection in [18]. A mapreduce thread model is designed to parallel and speed up the observation steps. Similarly, a multi-cue-based face-tracking algorithm with the supporting framework using parallel multi-core and one Graphic Processing Unit (GPU) is proposed in [19]. The multi-core parallel scheme can increase the speed by 2 - 6 times compared with that of the corresponding sequential algorithms.

### 3. Approach

#### Particle Filter

Particle filter is a robust tracking technique for moving objects in a cluttered environment. It is used to detect and track moving objects in the manner of non-linear and non-Gaussian problems. The belief distribution over the location of a tracked object is represented by multiple discrete “particles”.

Particle filter is one of Bayesian sequential importance sampling techniques. A finite set of weighted samples is used to recursively approximate the posterior distribution. Prediction and update are two essential steps in particle filter. Given all the observations up to time  $t-1$ , denotes as  $z_{1:t-1} = z_1, \dots, z_{t-1}$ , the probabilistic system transition model  $p(x_t | x_{t-1})$  is used in the prediction stage to predict the posterior at time  $t$ :

$$p(x_t | z_{1:t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1} \quad (1)$$

The state at time  $t$  with an observation  $z_t$  is updated using Bayes' rule:

$$p(x_t | z_{1:t}) = \frac{p(z_t | x_t) p(x_t | z_{1:t-1})}{p(z_t | z_{1:t-1})} \quad (2)$$

where  $p(z_t | x_t)$  is described by the observation equation. A finite set of  $N$  samples  $X_t = x_t^1, \dots, x_t^N$  with importance weight  $W_t = w_t^1, \dots, w_t^N$  is used to recursively approximate the posterior in Equation (2). The weight of each sample is calculated as:

$$w_t^j = w_{t-1}^j \frac{p(z_t | x_t^j) p(x_t^j | x_{t-1}^j)}{p(x_t | x_{t-1}, z_{1:t})} \quad (3)$$

In order to avoid degeneracy, the importance weights are used to resample the samples and an unweighted particle set is generated. The procedure of proposed particle filter for single object tracking is listed in Algorithm 1.

---

**Algorithm 1:** Particle Filter for Single Object Tracking

---

```

1 procedure Main;
2 initialization;
3 initial particle location matrix  $P_1$ 
4 foreach  $i \in P_1$  do
5   |  $i \leftarrow \text{random}(\text{height}, \text{width})$ 
6 end
7 procedure Map;
   Input : a set of consecutive frame  $X$ , a set of particle locations  $P_i, i \in \{2, \dots, n\}$  and Interference
           parameters:  $p, v$  and  $c$ 
   Output: new sets of particles  $P_i'$  and a new set of detected frame  $X'$ 
8 for  $i \leftarrow 2$  to  $P_n$  do
9   |  $P_i \leftarrow \text{updateParticles}(P_{i-1}, p, v)$ ;
10  | for  $j \leftarrow 1$  to  $P_i$  do
11  |   |  $C_j \leftarrow \text{getRGB}(P_i)$ ;
12  |   |  $W_i \leftarrow \text{calWeight}(C_j, c)$ ;
13  | end
14  |  $P_i \leftarrow \text{resampling}(W, P_i)$ ;
15  |  $X^i \leftarrow \text{drawParticles}(X_i, P_i)$ ;
16 end

```

---

Generally, a set of particles will be drawn uniformly for the initialization. The particles will be positioned from the proposal distribution in order to find the features of the moving object. The weight of the particles will be updated and it will then be normalized. The mean position of the particles will be calculated as the output of each iteration used the posterior distribution. The particles will be resampled in the next iteration with probability. The whole process will converge the particles in a position to track the moving object. In order to parallelize the steps of particle filter, the proposed algorithm divided the aforementioned steps into a map step and a reduce step.

## 4. Experiments and Results

### 4.1. Environmental Setup

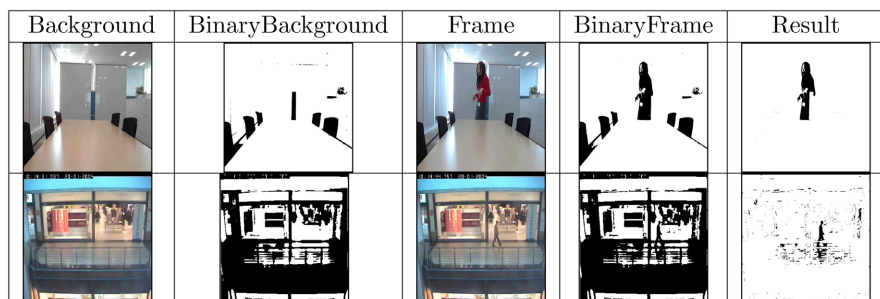
The experiments are conducted on an Apache Hadoop cluster with two data nodes using MapReduce. The two data nodes are setting in a virtual machine using VMware workstation pro 12.5 on a Windows machine (Intel(R) Core(TM) i7-6700K CPU 4.00 GHz 4.01 GHz). The video set is adopted from [20] and [21]. The first video has a lady in red walks in and out a room. The second video is a

person in gray walking in the hallway of a shopping mall. The experimental environments are adopted and modified from [22]-[25].

## 4.2. Background Subtraction

In order to detect a dynamic object from a video sequence, the background subtraction is a popular technique which is used for foreground segmentation by comparing current frame with the background image. The background images need to be preprocessed by image binarization [26]. An adaptive threshold is applied to binarize the images. The threshold is calculated by Otsu's method which is used to reduce the gray-level of image to a binary image [27].

In **Figure 1**, the background images are listed in the first column. The second column is the binary images after image binarization. The third column includes the sample frames in the two videos. The fourth column lists the binary images and the background subtraction results are demonstrated in the last column.



**Figure 1.** Background subtraction.

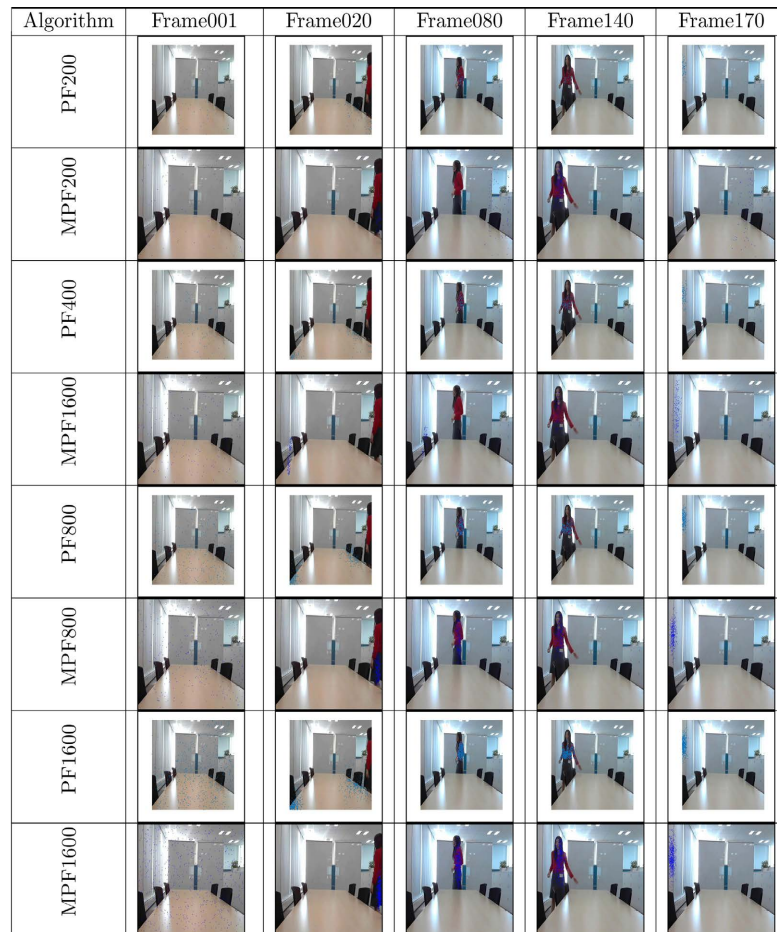
## 4.3. Tracking Results

In order to evaluate the tracking result, the proposed PF with MapReduce (MPF) is compared with the conventional particle filter (PF) using 200, 400, 800 and 1600 particles, respectively in **Figure 2**. The PF calculates the color likelihood which is tracking the red color of the person. However, the proposed MFP can track the person video with better accuracy and it's not limited to red color. Moreover, the MPF can converge much faster than CPF.

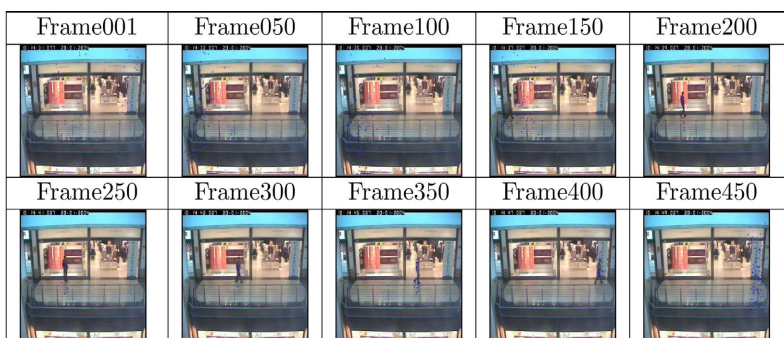
Furthermore, the tracking result of the proposed MPF applied to a shopping mall video using 200 particles is listed in **Figure 3**. The result of 400 particles is also illustrated in **Figure 4**, The result of 800 particles is also illustrated in **Figure 5**, The result of 1600 particles is also illustrated in **Figure 6**. The results indicate that a large number of particles can converge much faster and provide better accuracy as compared to a smaller number of particles.

## 4.4. Execution Time

Another important performance measurement for object tracking is the execution time. In **Figure 7**, the execution time of the proposed algorithm is estimated on a single-node Hadoop cluster and a two-node clusters. The experiment is conducted with the number of particles varying from 800 to 12,800. The proposed



**Figure 2.** A comparison of conventional particle filter (CPF) and particle filter with MapReduce (MPF) using 200, 400, 800 and 1600 particles, respectively.



**Figure 3.** A list of frames using MPF with 200 particles.

algorithm can process 12 frames per second in a single-node machine. It then increases to 13 frames per second for two-node cluster. The overall performance indicates that two-node cluster is faster than the single-node cluster. Furthermore, the proposed algorithm is conducted on multiple-nodes cluster from 1 to 16 nodes. The minimum, average, and maximum execution time are compared in **Figure 8**. As the number of clusters increases, the communication overhead increases. The minimum and average execution time are converged.

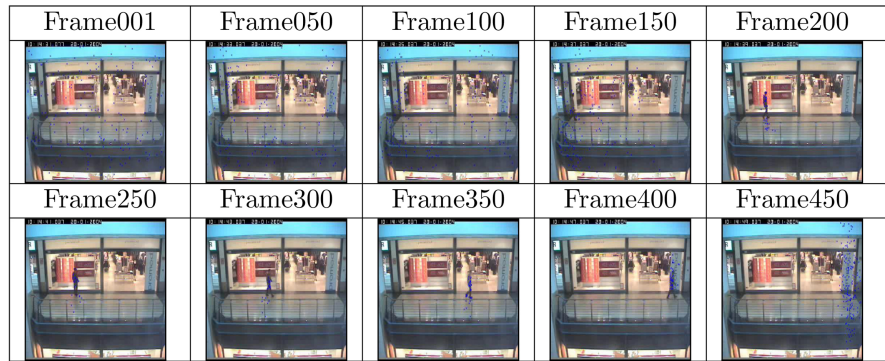


Figure 4. A list of frames using MPF with 400 particles.

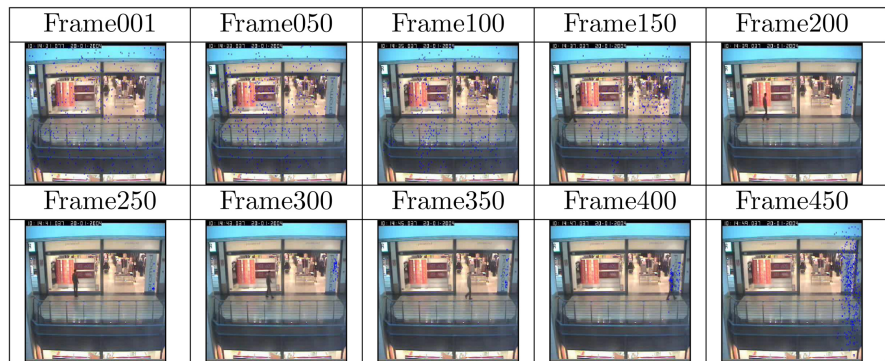


Figure 5. A list of frames using MPF with 800 particles.

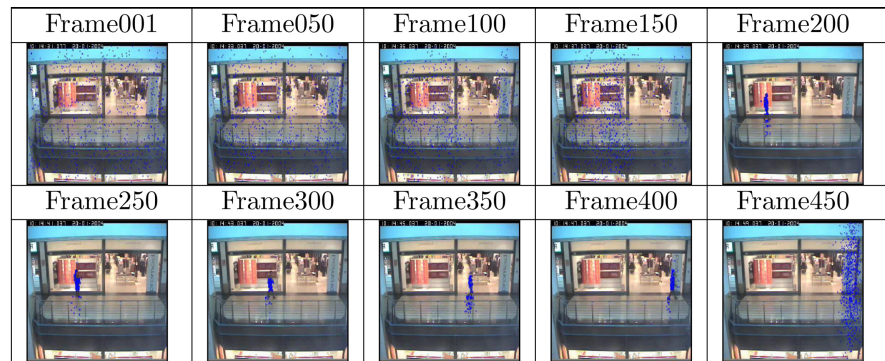


Figure 6. A list of frames using MPF with 1600 particles.

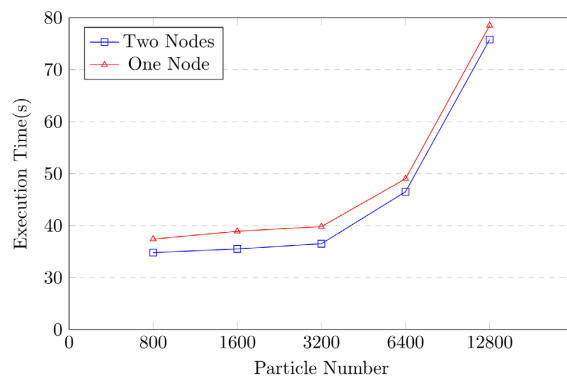
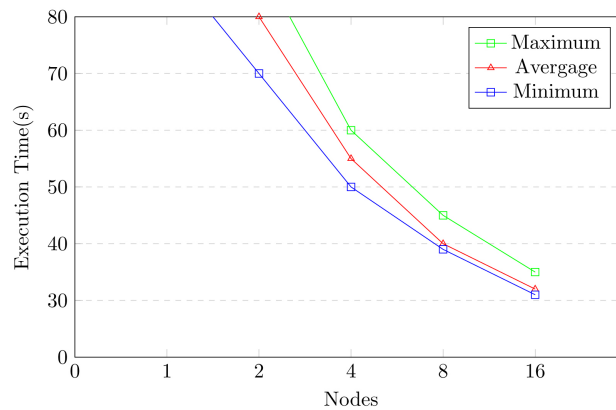


Figure 7. Execution time comparison.



**Figure 8.** Execution time comparison.

## 5. Conclusion

This paper first reviewed the current state of visual object trackers with connection to distributed system, then discussed past research on the application of the particle filter applied to object tracking and investigated their drawbacks. A novel parallelized particle filter has been proposed to overcome the problems identified by previous approaches, where the MapReduce mechanism is used to explore the scalability. Experiments and results demonstrated that the proposed method (MPF) does improve the tracking performance without introducing extra computational cost. Overall, it is shown that the proposed MapReduce-based particle filter works quite well together with the Hadoop distributed system. It is also demonstrated that the proposed algorithm is an efficient yet effective method applied to object tracking. In the future, the proposed algorithm will be improved to do multiple-object tracking and some other edge detection techniques can be adopted to improve the performance of the proposed MPF.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Yilmaz, A., Javed, O. and Shah, M. (2006) Object Tracking. *ACM Computing Surveys*, **38**, Article No. 13. <https://doi.org/10.1145/1177352.1177355>
- [2] Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A. and Hengel, A.V.D. (2013) A Survey of Appearance Models in Visual Object Tracking. *ACM Transactions on Intelligent Systems and Technology*, **4**, Article No. 58. <https://doi.org/10.1145/2508037.2508039>
- [3] Xia, G. and Ludwig, S.A. (2016) Object-Tracking Based on Particle Filter Using Particle Swarm Optimization with Density Estimation. 2016 *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, 24-29 July 2016, 4151-4158. <https://doi.org/10.1109/cec.2016.7744317>
- [4] Yang, C., Davis, L. and Duraiswami, R. (2005) Fast Multiple Object Tracking via a Hierarchical Particle Filter. *10th IEEE International Conference on Computer Vi-*

- tion (ICCV'05), Volume 1, 212-219.
- [5] Doucet, A., de Freitas, N. and Gordon, N. (2001) Sequential Monte Carlo Methods in Practice. Springer-Verlag.
  - [6] Jepson, A.D., Fleet, D.J. and El-Maraghi, T.F. (2003) Robust Online Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 1296-1311. <https://doi.org/10.1109/tpami.2003.1233903>
  - [7] Wang, D. and Chen, M. (2017) Particle Filter in Single Object Tracking with MapReduce. *Proceeding of 17th Industrial Conference on Data Mining (ICDM 2017)*, Newark, 12-16 July 2017, 11-19.
  - [8] Viola, P. and Jones, M.J. (2004) Robust Real-Time Face Detection. *International Journal of Computer Vision*, **57**, 137-154. <https://doi.org/10.1023/b:visi.0000013087.49260.fb>
  - [9] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, 886-893. <https://doi.org/10.1109/cvpr.2005.177>
  - [10] Felzenszwalb, P., McAllester, D. and Ramanan, D. (2008) A Discriminatively Trained, Multiscale, Deformable Part Model. 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 23-28 June 2008, 1-8. <https://doi.org/10.1109/cvpr.2008.4587597>
  - [11] van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T. and Smeulders, A.W.M. (2011) Segmentation as Selective Search for Object Recognition. 2011 *International Conference on Computer Vision*, Barcelona, 6-13 November 2011, 1879-1886. <https://doi.org/10.1109/iccv.2011.6126456>
  - [12] He, K., Zhang, X., Ren, S. and Sun, J. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. <https://doi.org/10.1109/tpami.2015.2389824>
  - [13] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision (ICCV)*, Santiago, 7-13 December 2015, 1440-1448. <https://doi.org/10.1109/iccv.2015.169>
  - [14] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 779-788. <https://doi.org/10.1109/cvpr.2016.91>
  - [15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., *et al.* (2016) SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV 2016*, Springer International Publishing, 21-37. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
  - [16] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017) Feature Pyramid Networks for Object Detection. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 2117-2125. <https://doi.org/10.1109/cvpr.2017.106>
  - [17] Yang, C., Duraiswami, R. and Davis, L. (2005) Fast Multiple Object Tracking via a Hierarchical Particle Filter. *10th IEEE International Conference on Computer Vision (ICCV'05)*, **1**, 212-219.
  - [18] Liu, K.Y., Li, S.Q., Tang, L., Wang, L. and Liu, W. (2009) Fast Face Tracking Using Parallel Particle Filter Algorithm. 2009 *IEEE International Conference on Multimedia and Expo*, New York, 28 June-3 July 2009, 1302-1305.

- 
- [19] Liu, K., Li, Y., Li, S., Tang, L. and Wang, L. (2011) A New Parallel Particle Filter Face Tracking Method Based on Heterogeneous System. *Journal of Real-Time Image Processing*, **7**, 153-163. <https://doi.org/10.1007/s11554-011-0225-6>
- [20] MathWorks (2016) Image Acquisition Toolbox: MATLAB 7.13 (R2011b).
- [21] EC Funded CAVIAR Project, EC Funded CAVIAR Project/IST 2001 37540. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>
- [22] Ma, H., Tang, H., Zhang, H., Zhao, X. and Kou, Y. (2015) HVPI: Extending Hadoop to Support Video Analytic Applications. 2015 *IEEE 8th International Conference on Cloud Computing (CLOUD)*, New York, 27 June-2 July 2015, 789-796.
- [23] Gordon, R. (1998) Essential JNI: Java Native Interface. Prentice Hall, Inc.
- [24] Liang, S. (1999) The Java Native Interface: Programmer's Guide and Specification. Addison-Wesley Professional.
- [25] Tan, H.L. and Chen, L.D. (2014) An Approach for Fast and Parallel Video Processing on Apache Hadoop Clusters. 2014 *IEEE International Conference on Multimedia and Expo (ICME)*, Chengdu, 14-18 July 2014, 1-6.
- [26] Tsai, D.-M. and Lai, S.-C. (2009) Independent Component Analysis-Based Background Subtraction for Indoor Surveillance. *IEEE Transactions on Image Processing*, **18**, 158-167. <https://doi.org/10.1109/tip.2008.2007558>
- [27] Sankur, B. (2004) Survey over Image Thresholding Techniques and Quantitative Performance Evaluation. *Journal of Electronic Imaging*, **13**, 146-165. <https://doi.org/10.1117/1.1631315>