Turkish Journal of Electrical Engineering and Computer Sciences

Volume 27 | Number 3 Article 13

1-1-2019

Low-cost multiple object tracking for embedded vision applications

MUHAMMAD IMRAN SHEHZAD

FAZAL WAHAB KARAM

SHOAIB AZMAT

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 1737 – 1751

doi:10.3906/elk-1807-1

Low-cost multiple object tracking for embedded vision applications

M. Imran SHEHZAD*, **Fazal Wahab KARAM, Shoaib AZMAT**
Department of Electrical Engineering, COMSATS University, Abbottabad, Pakistan

Received: 05.07.2018 • **Accepted/Published Online:** 18.01.2019 • **Final Version:** 15.05.2019

Abstract: This paper presents a low-cost multiple object tracking (MOT) technique by employing a novel appearance update model for object appearance modeling using K-means. The state-of-the-art work has attained a very high accuracy without considering the real-time aspects necessitated by currently trending embedded vision platforms. The major research on multiple object tracking is used to update the appearance model in every frame while discounting its persistent nature. The proposed appearance update model reduces the computational cost of the state-of-the-art MOT 6-fold by exploiting this facet of persistent appearance over the sequence of frames. To ensure accuracy, the proposed model is tested on different publicly available standard datasets with challenging situations for both indoor and outdoor scenarios. The experimental results illustrate that our model successfully achieves multiple object tracking while coping with long-term and complete occlusion. The proposed method achieves the same accuracy in comparison with the state-of-the-art baseline methods. Moreover, and most importantly, the proposed method is cost-effective in terms of computing and/or memory requirements in comparison to the state-of-the-art techniques. All these traits make our design very suitable for real-time and embedded video surveillance applications with low computing/memory resources.

Key words: Appearance update model, histogram, k-means, Gaussian mixture model, multiple object tracking, occlusion

1. Introduction

Multiple object tracking (MOT) is the process of estimating the state of moving objects in a sequence of video frames captured using a camera. It is a method of segmenting and observing an object's spatiotemporal modifications, i.e. to keep track of its presence, orientation, shape, motion, position, size, and occlusion, and to extract context information, which is useful for higher-level applications. MOT is a primary issue in computer vision, which has extensive applications in diverse video investigation scenarios. This significance is inspired by various applications, including video surveillance systems [1], human-machine interaction [2], vision-based robot navigation [3], sports analysis, and autonomous driving. In addition to the MOT, these applications may include, as part of their working, higher level tasks such as object recognition, activity investigation, and high-level event comprehension. MOT can be a time-demanding task due to the complex operations it performs on the large quantity of data in a video sequence. Precise and real-time MOT can significantly enhance the performance of higher level tasks of object recognition, activity investigation, and event comprehension [4–6].

Development of MOT algorithms is constrained by object/target detection because of background variations [7]. Generally, background subtraction and/or object detection classifiers are used for detection purposes. In recent years, due to the development of state-of-the-art object detection classifiers, the performance of object

*Correspondence: imranshahzad@ciit.net.pk

tracking methods has shown significant improvements [8, 9]. The most commonly used baseline customized detector for pedestrians was proposed by Dalal et al. [10]. They used the histogram of oriented gradients approach for feature representation and SVM as a classifier. This algorithm is used in many MOT algorithms for pedestrian detection [11, 12]. The other prominent baseline algorithm, AdaBoost, was proposed by Viola Jones, which cascades different weak classifiers to detect objects. Jongseok et al. [13] combined motion and color information for pedestrian detection and tracking using the AdaBoost algorithm. The recurrent neural network (RNN) was employed in [14, 15] for the classification of detection responses and association during tracking. The RNN network forms the connections between nodes in different frames to create a directed graph over a sequence to generate the trajectories. Convolutional neural networks (CNNs) have been recently used for multiple object detection and tracking purposes [16, 17]. The CNN is a class of deep feedforward artificial neural networks, trending classifiers for MOT and other computer vision applications due to comparatively less preprocessing requirements compared to other image classification algorithms. Although the customized detection classifiers discussed above offer good accuracy, they limit the performance of the MOT algorithms due to inherent computational costs. In addition, customized detector-based methods are not general object detectors, i.e. at a time they are trained to detect only one type of object, which in the above case is human. Moreover, customized detectors require extensive training to make them work for different datasets [9]. Despite extensive training, these detectors are inherently more biased towards training datasets. Regardless of the huge amount of research with high accuracy, a sample of which we have seen above, resource-efficient solutions to MOT problems is the least explored area. Many state-of-the-art proposals have achieved good accuracy, but they require more computational and memory resources. Consequently, these methods are not suitable for real-time embedded vision applications with low resources.

A few designs have achieved real-time computational requirements with the desired accuracy. Nguyen et al. [18] developed a template matching method for multiple object tracking by applying Kalman filters to model appearance features. However, the performance of this type of representation is easily constrained by illumination variations and occlusion [19]. Kratz et al. [20] employed color histograms to model appearance; they used a histogram distance-based probability function to achieve multiple object tracking. Yang [21] also utilized histograms for object appearance modeling to track multiple objects by analyzing histograms. The aforementioned histogram-based tracking methodologies require large memory despite their low computational cost. These memory-intensive algorithms might be troublesome in developing real-time embedded surveillance systems like smart cameras having low memory resources.

Numerous researchers have employed the Gaussian mixture model (GMM) and expectation maximization (EM) for appearance modeling to achieve MOT with low memory requirements. Henriques [22] proposed a covariance matrix descriptor to compute the probability of detection responses by applying Gaussian distribution on corresponding appearance models. This method requires prior modeling of numerous components and hence results in higher computational overhead. Tao et al. [23] gave a proposal based on motion, appearance, and shape models to accomplish highly precise MOT by employing an EM algorithm. This method has fairly high computational complexity as it requires estimation of different model components and parameters. Khan et al. [24] used the GMM for object modeling both in color and spatial domains to achieve MOT. This model estimates five-dimensional Gaussians to represent an object model. Papadourakis et al. [25] proposed a GMM-based appearance model along with an ellipse-based shape model for multiple object tracking by exploiting the principle of object permanence to deal with occlusions. All of the above-mentioned methods achieve high accuracy with low memory requirements. However, the underlying soft assignment nature of the baseline GMM

method and the periodic update of the object's appearance model in every frame results in high computational cost.

In our previous work [26], we designed a novel K-means-based appearance model for MOT to address both the issues of memory requirements and computational cost. It models the object's appearance based on K-means while presenting a new statistical distance metric for object association after occlusion. The K-means-based method is used to bridge the performance gap of baseline GMM and histogram methods [26]. The K-means-based method outperforms the GMM method in terms of computational time with comparable accuracy while having similar memory requirements. The K-means method also outperforms the histogram method in terms of memory requirements while achieving comparable accuracy. However, the iterative process to model object appearance using K-means in each frame makes it computationally expensive as compared to histogram-based methods, which can bar it from real-time applications. Moreover, the computational complexity is also due to the fact that random initialization of K-means centroids takes considerable time to converge.

The major research on MOT is primarily focused on attaining high accuracy with minimum errors, rather than the practical aspect of realizing a method with less computational time to meet real-time requirements [27, 28]. Therefore, resource utilization has not been properly addressed and is one of the major problems for the development of MOT-based analytic algorithms aiming at embedded vision platforms. The aforementioned MOT techniques summarize the problems pertaining to the existing models, i.e. memory and processing requirements. Few research efforts have been made to address this issue of resource utilization with comparable accuracy [29–31]. One of the major aspects of all of the above algorithms in the literature is that the appearance modeling in these algorithms is updated in every frame by disregarding the persistent nature of appearance. This is the one of the major sources that supplements the performance penalty in terms of computational cost in addition to the computational load of the underlying object initialization scheme, appearance model, and tracking strategy.

In this work, we propose a novel appearance update model to reduce the frequency of appearance modeling, as appearance normally tends not to change abruptly due to its persistent nature. Furthermore, histogram-based centroid initialization is employed to reduce convergence delay. The proposed appearance update model reduces the computational cost manifolds to make our algorithm suitable for real-time applications. Our main contributions and findings in this work are listed below:

- We present a novel object appearance update model for multiple object tracking and the proposed model can be applied to any MOT technique (e.g., histogram-based MOT, GM- based MOT, or K-means-based MOT) for significant reduction in computational cost.
- We have applied our algorithm to K-means-based multiple object tracking [26], in our previous work, to show its manifold performance gain. We have also incorporated histogram-based centroid initialization to reduce the convergence time of K-means.
- Our proposed method successfully achieved MOT in different indoor and outdoor scenarios in the presence of short/long-term and partial/complete occlusion with low appearance update rate. We validated our proposed algorithm on different standard datasets.
- The proposed design outperforms the K-means-based algorithm as it is more than 6 times faster. Moreover, the proposed method increases the memory requirements slightly as compared to the K-means-based algorithm. However, our design still requires 128 times less memory as compared to histogram-based techniques for the desired accuracy.

Table 1. Symbols and abbreviations with meanings.

Notation	Description
GMM	Gaussian mixture model
MOT	Multiple object tracking
obj	Object
rect	Rectangle, bounding box around an object
K	Appearance model
μ_i	Mean of the ith cluster
w_i	Weight of the ith cluster
$\mu_{i,t}$	Mean of the ith cluster in current iteration
$\mu_{i,t-1}$	Mean of the ith cluster in previous iteration
tot_obj	Total number of registered objects
O_i	ith object
tot_blobs	Total number of detected blobs in current frame
b_j	jth blob
$b_{(hist\_mag(i))}$	ith histogram peak of a blob
$O_{(hist\_mag(i))}$	ith histogram peak of an object
$b_{(bins(i))}$	Bin index of ith peak of a blob
$O_{(bins(i))}$	Bin index of ith peak of an object

The rest of the paper is organized as follows. Section 2 comprehensively explains the design methodology. Section 3 presents results and a brief discussion of the results. At the end, conclusions and /future research directions are summarized in Section 4. The symbols and abbreviations used in this paper are presented in Table 1.

2. Proposed methodology

The design flow of the proposed methodology is shown in Figure 1. The proposed technique extracts moving objects (blobs) using a background subtraction technique. The background modeling technique is based on multiple-model means as presented by Apewokin et al. [32]. The appearance model of a newly emerged blob is estimated along with it spatial position when seen for the first time, and it is registered as a new object. Afterwards, under no occlusion, the appearance model of the respective object's blob in the proceeding frames is updated only when there is a significant change in the appearance. On the other hand, the spatial position is updated in every frame. Similarly, when under occlusion, the appearance model is never updated while the spatial position of the occluded objects is updated to keep track of the objects. Subsequently, the objects are associated after the end of occlusion using the appearance model. Comprehensive details of the design are presented in subsequent sections.

Section 2.1 is about background subtraction and blob detection. Section 2.2 deals with object modeling based on its spatial location and visual appearance. Section 2.3 deals with blob/object registration and association based on its appearance model and spatial position estimated in the previous frame. Moreover, this section also demonstrates the use of the object appearance update model.

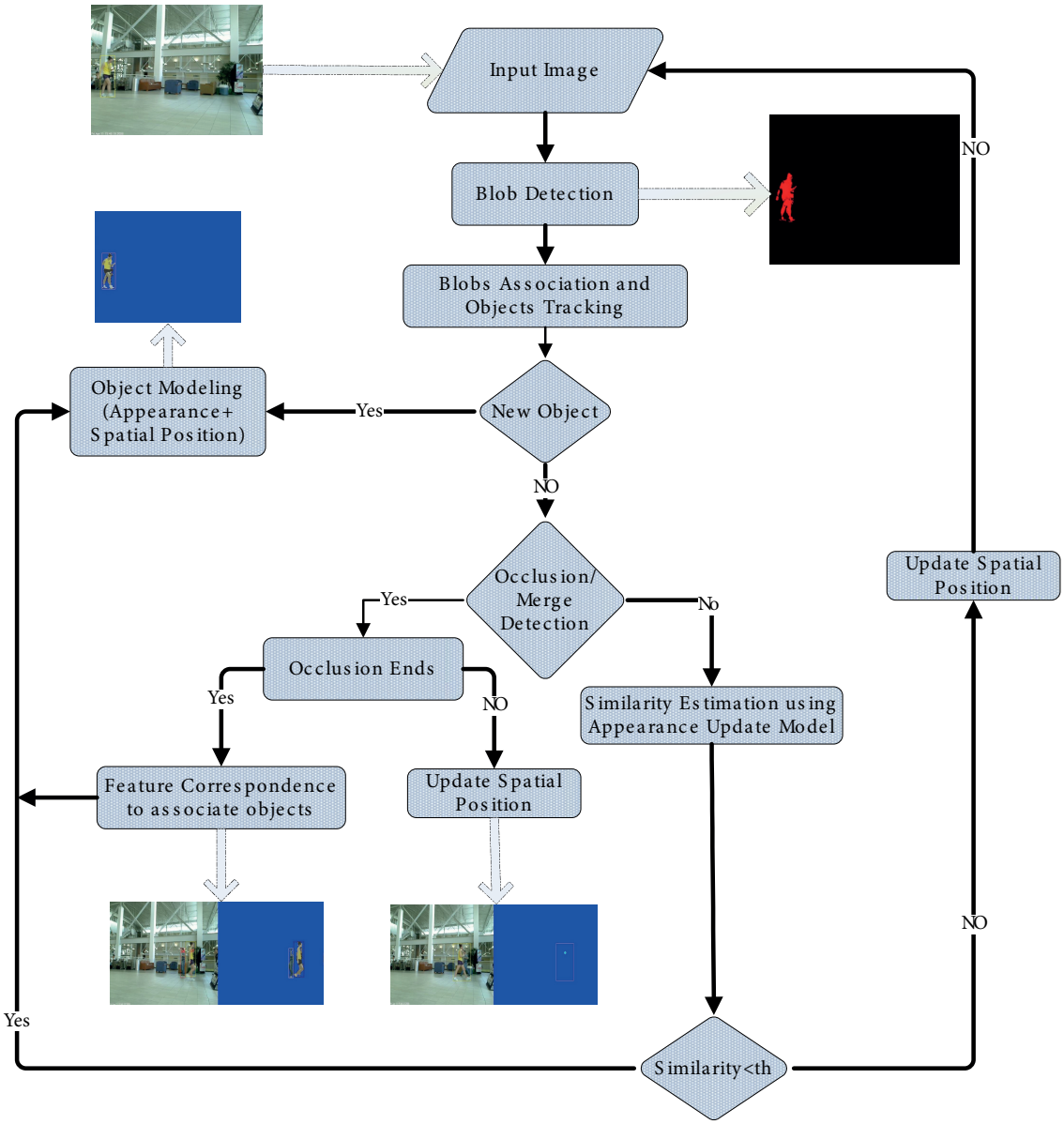

Figure 1. Overall design flow of the system with object detection, appearance modeling, and tracking.

2.1. Blob detection

At first, the dynamic background model is estimated using the multimodal mean [32], and then background subtraction is performed for foreground pixel extraction from each frame of the input video sequence. For a still camera, background subtraction generates a change mask for a potential moving object; the connected component analysis is further applied to find the distinct blob of each object. There is a bijection (one-to-one) mapping between blobs and objects, which means that every distinct object gives rise to exactly one blob. This bijection mapping is a fundamental assumption of the proposed method. Morphological operations are used to maintain this assumption while tracking. Moreover, the blob size filter is also employed to discard disconnected and undesired smaller blobs.

Algorithm 1 Appearance modeling algorithm.

Require: RGB color image, detected blob
Compute histogram of blob/object
Take top M peaks bins
for $i := 1$ to M : **do**
Find RGB triplet for ith respective bin from the pool of M chosen bins
end for
Initialize clusters centroids $\mu_1, \mu_2, \ldots\ldots\ldots \mu_M$ using respective RGB triplet of M bins
Begin: $temp_i = argmin\|I(x,y) - \mu_(i,t)\|^2$ where $i = 1$ to M
for $i := 1$ to M : **do**
$S_i =$ place $I(x,y)$ to nearest $temp_i$
end for
for $i := 1$ to M : **do**
$\mu_{i,t} = \frac{\sum_{I(x,y)\in S_i} I(x,y)}{cluster\_count_i}$
end for
Repeat Begin until $\mu_(i,j)$ converges
end

2.2. Object modeling

Object appearance is modeled using K-means as proposed in our previous work [26] with additional histogram-based centroid initialization [33]. Centroid initialization in K-means is one of the major challenges, and it needs to be addressed for fast convergence and appropriate clusters. In the proposed histogram-based initialization, the top M (bins) peaks of the color histogram are taken as initial cluster centroids. The object model is represented as obj(rect,K); the rectangle *rect* represents a spatial bounding box (position) around an object and K represents the visual appearance of an object.

The appearance model is represented as K(w_i, μ_i), $1 \leq i \leq M$, which denotes the color distribution of object pixels into various clusters with respective cluster centroid (mean) μ_i and its weight w_i, where M is the total number of clusters. Here, w_i, the weight of the ith cluster, is calculated as

$$w_i = \frac{cluster\_count_i}{total\_count}, \tag{1}$$

where cluster_count$_i$ represents total pixels in the ith cluster, while total_count is the total pixels in the blob/object. Finally, $\mu_{(i,t)}$, the mean of the color distribution of the ith cluster of the current iteration, is estimated using the K-means algorithm with initialization based on histogram peaks. We have used four clusters for appearance modeling, as four clusters are sufficient to appropriately model the pedestrians [26].

The K-means-based appearance model using the histogram-based initialization method is presented in Algorithm 1. The algorithm is divided into two stages, which are centroid initialization and estimation of final optimal centroids for clusters. Initially, the histogram of the extracted blob is calculated on the basis of RGB color channels. The top M bins of the histogram with respect to magnitude are chosen for initialization purposes.

$$z|(z \in hist_{blob}) \cap (z \in top_{bins}), \tag{2}$$

where z is the RGB index of the bin, while $hist_{blob}$ is the $16 \times 16 \times 16$ histogram of the blob and top_{bins} is the set of top M peaks of the histogram of the blob.

The top M chosen bins z are used to initialize centroids $\mu_{(i,t)}$ of the K-means clusters as these bins

suggest the color density distribution of the object. In this way, cluster centroids are initialized with the highly populated bin of the histograms to reduce the convergence time. Afterwards, it is the standard K-means algorithm, which places each pixel in the nearest cluster by measuring Euclidean distance. Subsequently, the algorithm recomputes the cluster centroids and repeats the process to find the optimal clusters such that

$$\mu_(i,t) = argmin \sum_{i=1}^{M} \sum_{I(x,y)\in S_i} \|I_{(x,y)-\mu_(i,t-1)}\|^2, \tag{3}$$

where $\mu_{(i,t-1)}$ is the mean of pixels in the S_i cluster of the preceding iteration, and I is the color value.

The rectangle model *rect(Xmax,Ymax,Xmin,Ymin)* represents the spatial boundary of an object, where *(Xmax,Ymax)* are the maximum and *(Xmin,Ymin)* are the minimum spatial coordinates. These coordinates are extracted in each frame for every object (blob); this spatial information is used for blob/object association to achieve MOT.

2.3. Blobs' association with objects/tracking

This section formulates the association of an extracted blob to an object. The newly observed blob will be registered as a new object in the respective frame, and then its association is made afterwards in the coming frames. We employ similar heuristics for object association/tracking as used in [25] and [26] with some changes. After blob detection, the following four scenarios may arise:

1. A blob has no association with existing objects.
2. An object is not associated with any extracted blob.
3. An object has association with only one blob.
4. A blob shows association with multiple objects.

2.3.1. A blob b has no association with any existing object

$$\forall O_i, b(rect) \cap O_i(rect) = \phi, \tag{4}$$

where $1 \leq i \leq tot\_obj$, and tot_obj is the total number of already registered objects.

This is evidence of a new object that has emerged in the scene for the first time, as it has not found any association with previously registered objects. Therefore, a new object will be registered and its appearance and spatial models are estimated.

2.3.2. An object O is not associated with any extracted blob

$$\forall b_j, b_j(rect) \cap O(rect) = \phi, \tag{5}$$

where $1 \leq j \leq tot\_blobs$, and tot_blobs is the number of extracted blobs in the current frame.

This shows that the object O does not have any association with extracted blobs, and it leads to the fact that object O has just left the scene. Therefore, the tracking of object O is stopped, as the respective object is no longer in the scene.

2.3.3. An object O has association with only one blob b

$$b(rect) \cap O(rect) \neq \phi. \tag{6}$$

This shows that a blob b has one-to-one correspondence with an object O, and hence the tracking identity is assigned to object O with its spatial model updated. As an object's appearance changes occasionally, there is no need to update the appearance model unless there is a significant change in its appearance. The proposed appearance update model exploits this fact by incorporating histogram similarity.

For the object appearance update model, on the registration of the newly emerged object, the top 8 peaks (with respect to magnitude) of the object color histogram are also stored along with the K-means object appearance model centroids. We have observed from our analysis of various standard datasets (mostly pedestrians) that the top 8 peaks are normally sufficient to indicate significant change in the appearance model. Therefore, in subsequent frames, the new histogram peaks of the corresponding blob of the object are compared with the respective stored histogram peaks for similarity estimation.

The proposed appearance update model is presented in Algorithm 2. First, the RGB color histogram of the extracted blob is computed. The top 8 bins of the histogram with respect to magnitude along with their bin indexes are chosen:

$$b_{hist\_mag} | (b_{hist\_mag} \in hist_{blob\_mag}) \cap (b_{hist\_mag} \in top_{8bins}), \tag{7}$$

where $b_(hist\_mag)$ is the magnitude of the bin, while $hist_(blob\_mag)$ is the magnitude of the $16 \times 16 \times 16$ histogram, and top_8bins is the set of top 8 peaks of the histogram:

$$b_{bins} | (b_{bins} \in hist_{blob}) \cap (b_{hist\_mag} \in top_{8bins}), \tag{8}$$

where b_{bins} is the RGB index of the bin, while $hist_{blob}$ is the $16 \times 16 \times 16$ histogram, and top_{8bins} is the set of top 8 peaks of the histogram:

$$dist = \sum_{1}^{8} |b_{hist\_mag(i)} \times b_{bins(i)} - O_{hist\_mag(i)} \times O_{bins(i)}|, \tag{9}$$

where $b_{(hist_mag(i))}$ is the magnitude of the ith bin for blob b and $b_{(bins(i))}$ is the bin index of the ith bin, while $O_{(hist_mag(i))}$ is the magnitude of the ith bin for object O and $O_{bins(i)}$ is the bin index of the ith bin. The K-means appearance model is updated when dist$>$threshold, and the top 8 peaks of the histogram are recomputed. On the contrary, the object model retains its previous K-means centroids and top 8 bins.

2.3.4. A blob b shows association with multiple objects

The one-to-one association of blobs and objects is defied during occlusion. Therefore, whenever multiple objects correspond to one blob, there is obviously a state of occlusion as multiple objects have merged to form a single blob. The trajectory of the merged blob is needed to be preserved throughout occlusion, and it is ensured by the update of blob rectangle coordinates in successive frames. The appearance model is not updated for the occluded objects during occlusion until they split. After the split of the occluding objects, their tracks are correctly reassigned by the feature correspondence method using the stored K-means appearance models of the objects as proposed in our previous work [26].

Algorithm 2 Appearance update model algorithm.

Require: RGB color image, top 8 bins of histogram

Compute histogram of blob b
Choose top 8 bins
for $i := 1$ to 8 : **do**
sort $b_{hist\_mag(i)}, b_{bins(i)}$
end for
Calculate Similarity
$dist = \sum_1^8 |b_{hist\_mag(i)} \times b_{bins(i)} - O_{hist\_mag(i)} \times O_{bins(i)}|$
if dist $>$ threshold **then**
Update appearance model using K-means algorithm
Update top 8 bins
else
Old appearance model will remain effective
end if

3. Experimental setup and results

The proposed technique is implemented using Microsoft Visual Studio C++ running on an Intel Core-2-Duo 2.10 GHz PC with 3 GB RAM. We have tested our design on the same standard datasets used in [26] to validate the performance of the proposed model. We evaluated the performance of the system for outdoor and indoor scenarios with occlusions. Table 2 provides the summary of datasets used to evaluate the proposed design. The results are illustrated below.

Table 2. Datasets.

Source	Number of frames	Frames per second	Resolution
CAVIAR Meet	826	10	320 × 480
Hall dataset	1462	3	640 × 480
PETS2001	2688	25	760 × 580
PETS 2009S2L1	794	-	760 × 580

The object appearance update model and tracking results are presented in Figure 2. When the appearance model is updated in a frame, it is shown using a set of clusters (different colors) inside the spatial bounding box, while the empty spatial bounding box is shown when the appearance model is not updated. The person in frame 300 has appearance modeled as shown in the blue frame, while frame 390 shows a case when the appearance update model does not remodel the appearance due to insignificant change in the appearance. The person's appearance has been updated only in 15 frames, from frame 300 to 390. This object appearance update in only 15 frames out of 91 frames makes the advantage of our appearance update model obvious. In frame 480, a second object (vehicle) has entered the scene for the first time and its appearance is modeled after registration of this object. The second object (vehicle)'s update rate is quite low as it is not updated until frame 640. This low update rate is because of its bigger size and uniform color, despite poor background subtraction due to shadows. On the other hand, the first object's update rate was comparatively higher, because of slightly poor background subtraction, very small visible size, and nonuniform color of the object. Our appearance update model successfully tracked both objects despite occlusion while updating the appearance model only when necessary, thus saving valuable computational time.

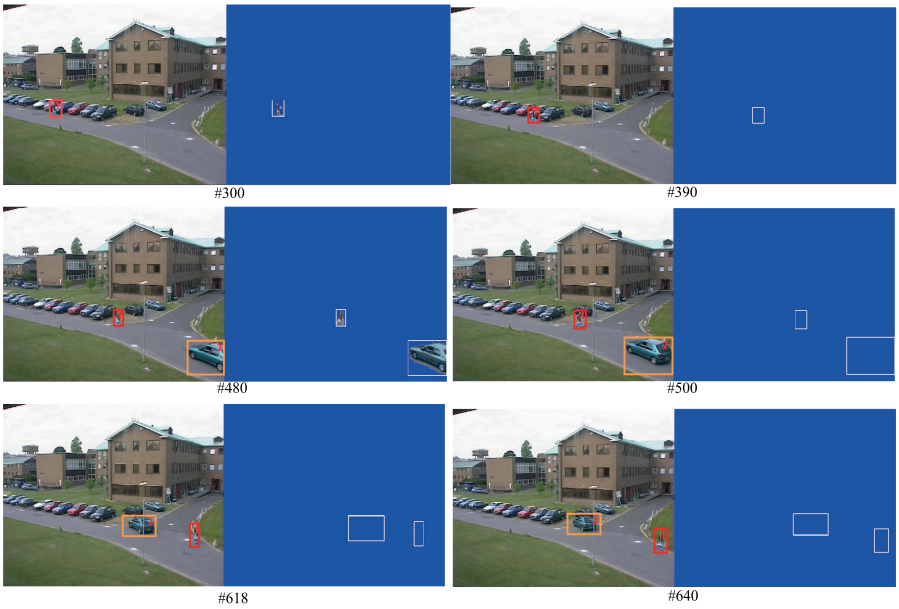

Figure 2. Appearance model update/tracking results.

The tracking results for different publicly available standard datasets are presented in Figure 3. The results of the CAVIAR dataset, where two people meet, interact, and leave each other after a few frames, are presented in Figure 3a. To illustrate the tracking of multiple objects, the objects are shown by different colored rectangles and their distinct IDs inside the rectangles. It is observed that different objects are tracked successfully despite long-term occlusion. The Hall dataset results are shown in Figure 3b, exhibiting a situation where objects move quickly while facing short-term occlusions. The results demonstrate the accuracy of the proposed model despite low frame rate. The results from PETS2001 are demonstrated in Figure 3c, dealing with complex scenarios of interacting objects having different sizes like humans and vehicles. The set of frames demonstrates the performance of the proposed method in the presence of different objects (pedestrians and vehicles). The last sequence is taken from PETS 2009S2L1, a widely used dataset for all MOT designs [26]. It involves pedestrians with high similarities in their appearances. The proposed algorithm achieved high accuracy despite high appearance similarities and frequent occlusion, as shown in Figure 3d. Our algorithm successfully handled all the complex cases of these publicly available standard datasets despite the proposed infrequent updates of the object appearance model. In summary, our algorithm achieves the same accuracy as the baseline algorithms of GMM, histogram, and K-means as it handles all complex scenarios of the standard datasets.

The most important contribution of this paper is presented in Table 3, i.e. the update rate for different datasets. The average update rate is defined as the average update of all the objects in the dataset, while the best-case update rate is defined as the lowest update rate required by an object among all the objects in the dataset, and the worst-case update rate is the highest update rate required by an object. The update rate depends on the quality of background subtraction, illumination changes, shadows, size of the visible object, and nonuniform color. The update rate of the PETS2009S2L1 dataset is very low because of uniformly sized objects, better blob detection, and controlled illumination conditions. The difference between worst and best-case update rates of the PETS2009S2L1 dataset is very low, suggesting the uniformity of update rate for all the objects, whereas the update rate of the Hall dataset is on the higher side because of poor blob detection and shadows. Furthermore, the frame rate of the Hall dataset is quite low and objects stay in the scene for a very

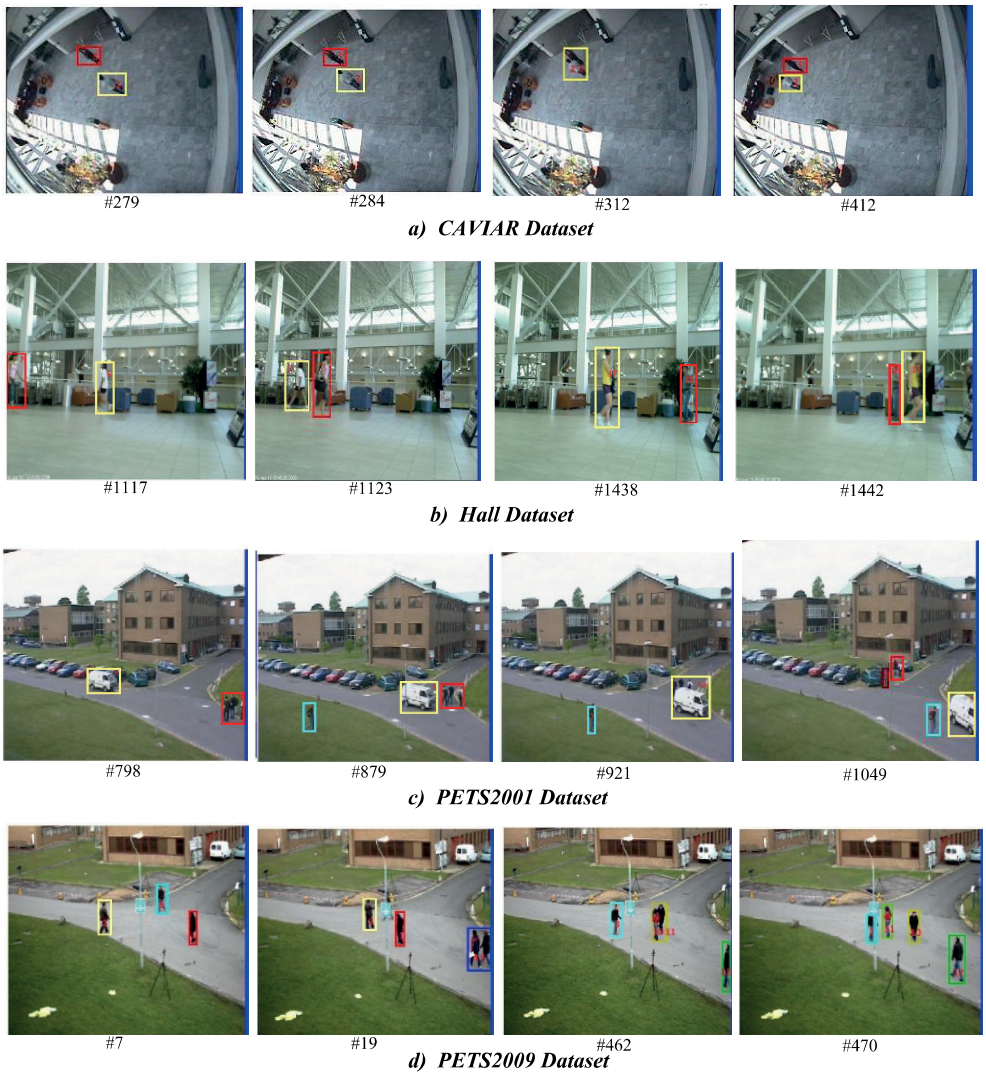

Figure 3. MOT results.

Table 3. Appearance model update rate.

Dataset	Average update rate	Best-case update rate	Worst-case update rate
CAVIAR Meet	18.18%	8%	27%
PETS2001	14.70%	2.27%	20%
Hall	26.08%	10%	40%
PETS 2009S2L1	9.25%	6.66%	12%

short time. This low appearance update, i.e. updating appearance only when necessary, results in manifold reduction in computational time as discussed in the following section.

3.1. Computational cost and memory requirements

We have compared the qualitative metrics of the proposed model with baseline and state-of-the-art trackers aimed at real-time embedded applications with respect to 'speed' and 'memory' as suggested in [34]. The average

execution time of the proposed algorithm for modeling a single object of different datasets is presented in Table 4. It is observed that the computational performance of the proposed method has significant improvement in

Table 4. Average computational time (ms).

Dataset	Proposed method	K-means-based method [26]	GMM [26]	Histogram bin size=(16 ×16×16) [26]	EAMTT [29]	GMPHD [30]	TSDA_OAL [31]
CAVIAR Meet	1.68	10	59	0.12	9.14	4.81	7.87
PETS2001	6.07	46	240	0.25	14.54	8.56	8.61
Hall	16.57	72	290	0.39	18.61	19.37	22.44
PETS 2009S2L1	4.17	49	262	0.27	15.39	9.10	9.15

comparison to the state-of-the-art trackers. For the majority of the trackers, the computationally expensive part is appearance modeling. Our proposed appearance update model significantly reduces the computational cost by exploiting the persistent nature of the object's appearance. The proposed algorithm is on the slower side as compared to the histogram-based method. However, the speed of the histogram-based methods is strongly influenced by the number of target objects, as it can cause extra memory retrieval time for object model acquisition from the main memory [35].

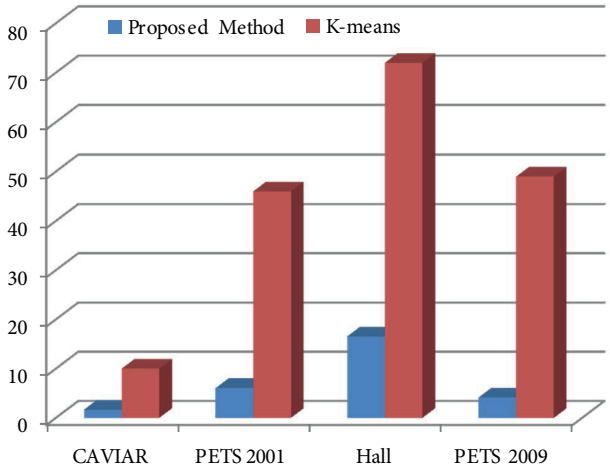

Figure 4. Performance improvement over [26].

.

The significant improvement of the proposed model in terms of computational cost for all the datasets in comparison with the K-means-based method [26] is presented graphically in Figure 4. Our design outperforms the K-means-based method for the PETS2009 and PETS2001 datasets by a quite large margin. This is due to the fact that many objects in these datasets stay in the scene for quite a long time without the need for updating their appearance model. Furthermore, these datasets have controlled illumination conditions and larger visible object areas. The proposed design also outperforms the other two datasets; however, the margin is a bit low. This lower performance is because the objects of the Hall dataset stay in the scene for a very short time and the blob detection is poor, whereas, for the CAVIAR dataset, the lower performance is because the results of background subtraction are not very good, and the object visible area is also very small.

Table 5. Memory consumption (bytes).

Proposed method, bytes/clusters	K-means-based method, bytes/clusters [26]	GMM, bytes/clusters [26]	Histogram, bytes/size of bins(R×G×B) [26]
28/3	12/3	21/3	512/8×8×8
32/4	16/4	28/4	4096/16×16×16
36/5	20/5	35/5	32768/32×32×32

The memory/storage requirements are compared in Table 5. The proposed model consumes almost the same memory as required by the GMM, and far less than that required by a histogram.

The proposed model has little increase in memory requirements as compared to the K-means-based method. This is because it needs to store top 8 bins magnitudes of the histogram and their corresponding bin indexes in addition to K-means centroids and their respective weights. For four clusters, the proposed technique requires storage of 4 RGB centroids, i.e. 12 bytes, 4 bytes to store corresponding weights of the centroids, and another 16 bytes to store histogram weights and bin indexes. Therefore, the total storage requirement for the proposed method is 32 bytes/4 clusters compared to 16 bytes/4 clusters required by the original K-means algorithm [26].

The proposed method bridges the performance gap of our previously proposed K-means-based method and histogram-based method. Our design is computationally quite fast as compared to the K-means and GMM-based methods with comparable memory requirements. Furthermore, our method is much better in terms of memory requirements as compared to the histogram-based method. All of these traits make our design suitable for real-time platforms with limited resources like embedded smart cameras.

4. Conclusion and future work

A low-cost MOT approach is presented in this paper. The proposed appearance update model reduced the computational cost manifolds by exploiting the persistent nature of appearance over the sequence of frames. Moreover, the intelligent cluster centroid initialization has further reduced the computational cost by decreasing the number of iterations. Our model has achieved the same accuracy as the baseline appearance-based MOT algorithms of GMM, histogram, and K-means for standard datasets. Moreover, the qualitative evaluation shows that the proposed model provides a cost-effective solution both in terms of memory and computational resources in comparison to the existing state-of-the-art baseline techniques. All these features make our design very useful for real-time applications, especially the ones using low-cost embedded video surveillance platforms with low computational and memory resources like smart cameras. It is important to note here that our appearance update model is generic in nature and it can be applied to any appearance model. Therefore, it can be applied to the histogram-based method or the GMM-based method with expected similar manifold reduction in computational time as for K-means. In the future, different techniques will be explored to make the K-means appearance model more dynamic by fair selection of number of clusters rather than a static approach. Furthermore, parallel architecture platforms will be employed to reduce the computational time.

References

[1] Ali I, Dailey MN. Multiple human tracking in high-density crowds. Image and Vision Computing 2012; 30 (12): 966-977.

[2] Hongyong T, Youling Y. Finger tracking and gesture recognition with kinect. In: IEEE 2012 International Conference on Computer and Information Technology; Chengdu, China; 2012. pp. 214-218.

[3] Benavidez P, Jamshidi M. Mobile robot navigation and target tracking system. In: IEEE 2011 International Conference on System of Systems Engineering; Albuquerque, NM, USA; 2011. pp. 299-304.

[4] Collins RT, Lipton AJ, Kanade T, Fujiyoshi H, Duggins D et al. A System for Video Surveillance and Monitoring. Pittsburgh, PA, USA: Carnegie Mellon University Press, 2000.

[5] Haritaoglu I, Harwood D, Davis LS. W/sup 4/: real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis & Machine Intelligence 2000; 22 (8): 809-830.

[6] Stauffer C, Grimson WE. Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis & Machine Intelligence 2000; 22(8): 747-757.

[7] Führ G, Jung CR. Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras. Pattern Recognition Letters 2014; 39: 11-20.

[8] Bae SH, Yoon KJ. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: IEEE 2014 Conference on Computer Vision and Pattern Recognition; Columbus, OH, USA; 2014. pp. 1218-1225.

[9] Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K. MOT16: A Benchmark for Multi-object Tracking. Ithaca, NY, USA: Cornell University Repository, 2016.

[10] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: IEEE 2005 Conference on Computer Vision and Pattern Recognition; San Diego, CA, USA; 2005. pp. 886-893.

[11] Breitenstein MD, Reichlin F, Leibe B, Koller-Meier E, Van Gool L. Robust tracking-by-detection using a detector confidence particle filter. In: IEEE 2009 International Conference on Computer Vision; Kyoto, Japan; 2009. pp. 1515-1522.

[12] Gaikwad V, Lokhande S. Vision based pedestrian detection for advanced driver assistance. Procedia Computer Science 2015; 46: 321-8.

[13] Lim J, Kim W. Detecting and tracking of multiple pedestrians using motion, color information and the AdaBoost algorithm. Multimedia Tools and Applications 2013; 65 (1): 161-79.

[14] Milan A, Rezatofighi SH, Dick AR, Reid ID, Schindler K. Online multi-target tracking using recurrent neural networks. In: AAAI 2017 Conference on Artificial Intelligence; San Francisco, CA, USA; 2017. pp. 4225–4232.

[15] Sadeghian A, Alahi A, Savarese S. Tracking the untrackable: learning to track multiple cues with long-term dependencies. In: IEEE 2017 International Conference on Computer Vision; Venice, Italy; 2017. pp. 300–311.

[16] Gan W, Wang S, Lei X, Lee MS, Kuo CC. Online CNN-based multiple object tracking with enhanced model updates and identity association. Signal Processing Image Communication 2018; 66: 95-102.

[17] Chu Q, Ouyang W, Li H, Wang X, Liu B et al. Online multi-object tracking using CNN based single object tracker with spatial-temporal attention mechanism. In: IEEE 2017 International Conference on Computer Vision; Venice, Italy; 2017. pp. 4846-4855.

[18] Nguyen HT, Smeulders AW. Fast occluded object tracking by a robust appearance filter. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004; 26 (8): 1099-1104.

[19] Luo W, Xing J, Milan A, Zhang X, Liu W et al. Multiple Object Tracking: A Literature Review. Ithaca, NY, USA: Cornell University Repository, 2014.

[20] Kratz L, Nishino K. Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 2012; 34 (5): 987-1002.

[21] Yang T, Pan Q, Li J, Li S. Real-time multiple objects tracking with occlusion handling in dynamic scenes. In: IEEE 2005 International Conference on Computer Vision and Pattern Recognition; San Diego, CA, USA; 2005. pp. 970-975.

[22] Henriques JF, Caseiro R, Batista J. Globally optimal solution to multi-object tracking with merged measurements. In: IEEE 2011 International Conference on Computer Vision and Pattern Recognition; Barcelona, Spain; 2011. pp. 2470-2477.

[23] Tao H, Sawhney HS, Kumar R. Object tracking with Bayesian estimation of dynamic layer representations. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002; 24 (1): 75-89.

[24] Khan S, Shah M. Tracking people in presence of occlusion. In: Springer 2000 Asian Conference on Computer Vision; Taipei, Taiwan; 2000. pp. 1132-1137.

[25] Papadourakis V, Argyros A. Multiple objects tracking in the presence of long-term occlusions. Computer Vision and Image Understanding 2010; 114 (7): 835-846.

[26] Shehzad MI, Shah YA, Mehmood Z, Malik AW, Azmat S. K-means based multiple objects tracking with long-term occlusion handling. IET Computer Vision 2017; 11 (1): 68-77.

[27] Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K. Motchallenge 2015: Towards a Benchmark for Multi-target Tracking. Ithaca, NY, USA: Cornell University Repository, 2015.

[28] Pandey M, Ubhi JS, Raju KS. Computational acceleration of real-time kernel-based tracking system. Journal of Circuits, Systems and Computers 2016; 25 (4): 1-30.

[29] Sanchez-Matilla R, Poiesi F, Cavallaro A. Online multi-target tracking with strong and weak detections. In: Springer 2016 European Conference on Computer Vision; Amsterdam, the Netherlands; 2016. pp. 84-99.

[30] Song YM, Jeon M. Online multiple object tracking with the hierarchically adopted gm-phd filter using motion and appearance. In: IEEE 2016 International Conference on Consumer Electronics-Asia; Seoul, South Korea; 2016. pp. 1-4.

[31] Ju J, Kim D, Ku B, Han DK, Ko H. Online multi-person tracking with two-stage data association and online appearance model learning. IET Computer Vision 2016; 11 (1): 87-95.

[32] Apewokin S, Valentine B, Wills L, Wills S, Gentile A. Multimodal mean adaptive background for embedded real-time video surveillance. In: IEEE 2007 Conference on Computer Vision and Pattern Recognition; Minneapolis, MN, USA; 2007. pp. 1-6.

[33] Tian M, Yang Q, Maier A, Schasiepen I, Maass N et al. Automatic histogram-based initialization of k-means clustering in CT. In: Springer 2013 Workshop Bildverarbeitung fur die Medizin; Heidelberg, Germany; 2013. pp. 277-282.

[34] Li X, Hu W, Shen C, Zhang Z, Dick A et al. A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology 2013; 4 (4): 1-48.

[35] Barry B, Brick C, Connor F, Donohoe D, Moloney D et al. Always-on vision processing unit for mobile applications. IEEE Micro 2015; 35 (2): 56-66.