Single and Multiple Object Tracking Algorithm Based on a Particle Grouping Approach with Occlusion Handling

D. Sudha  
Member, IEEE, Research Associate  
School of Computing Science and Engineering  
VIT University, Chennai, India  
E-mail: dsudha.89@gmail.com  
Tel: 07373832002

J.Priyadarshini  
Associate Professor, School of Computing Science and Engineering  
VIT University, Chennai, India  
E-mail: vinodha.nizanth@gmail.com  
Tel: 09486074882

Abstract

Multiple object detection and tracking is one of the hot research topics in the field of intelligent transportation systems, computer vision, robotics and medical image processing that is aptly fitting to solve the real time applications like traffic monitoring, occlusion handling, collision avoidance, lane changing assistance, public safety on the road and much more. When there is a huge necessity of detecting and tracking multiple objects in motion there exists a several challenges such as object shape, size, speed, memory, appearance, varying illumination, camera oscillations, cluttering, animals crossing on the road and shadowing etc. Ensuring that robustness and accuracy requires most optimized algorithms that aptly fit into the multiple object tracking problem. In this paper, we addressed a problem of single and multiple object tracking algorithm based on a particle grouping method and fusion of two optimal filters such as particle filter and Kalman filter for solving the linear and non-linear motion respectively, and also we addressed the other problem of various occlusion states such as full occlusions, semi occlusion, half occlusion and so on. The first step is for preprocessing which means system automation is to act intelligently in detecting the region of interest from a given input video by using haar trainer formerly known as viola Jones method paired with Adaptive Discrete Boost Classifier. In the experimental results we have shown that the training mechanism attains the accuracy of 85.9% detection rate of multiple objects in the presence of various occlusion states. The second step is used for particle generation and then next step follows particle grouping, trajectory estimations and tracking the multiple objects in a given input video of 20 frames per second. The final step towards the results of object detection engine, then the accuracy of the system has improved in comparison with that of the background subtraction and segmentation technique. Experimental results were implemented by using MATLAB 16.0.0a has proved that the particle grouping technique offered a 50% reduction in the number of particles that are required to be processed for tracking in comparison with that of particle filter alone. On an average the performance of the proposed system has increased by 40% in comparison to the existing systems. It can also be inferred that the
accuracy in case of occlusions has an increased by an average of 30% in comparison with the application of existing filters.

**Keywords:** Adaptive Boost Classifier, Back Propagation Network, Haar Training, Multi Layer Perceptron, Particle Grouping, Particle Filter, Occlusion handling

I. Introduction

Many decades ago, there was a huge demand for Multiple Object Detection and Tracking but still remains more challenging tasks exists in the field of Intelligent Transportation System, Computer Vision and endless higher level applications that requires both location and/or shape of the object in an every frame. [1] In general tracking rails with non-stationary object, target descriptions and background cluttering in which changes over time to time. Most of the available algorithms [2] and [3] are able to perform only tracking, in predefined and well controlled environments such that limitations exist to solve the non-linearity problem of the tracking system. The major issues were found when we detect and track multiple objects or vehicles on roads such as illumination variation, occlusion, camera oscillations and abrupt changes in motion.

The definition of tracking is defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. Real time based applications are explained in the field of computer vision and surveillance of multiple tracking based on motion recognition, vehicle navigation, automated surveillance and many more.

Detecting multiple moving objects on the road plays an important role in a wide range of applications in the field of intelligent transportation system, computer vision, medical coding, processing and robotics etc., so that multiple tracking brings most significant attention in recent years. However, multiple tracking of moving objects is still a challenging problem due to the major reasons such as abrupt object motion, changing appearance patterns of the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Error merges and object labeling [4-6] are other severe challenges in object tracking. Thus the proliferation of high-powered computers, inexpensive video cameras and sensors are increasing needs for system, automation of training and testing in an input video. Hence video analysis has generated a great deal of interest in object tracking algorithms.

In the existing literature reviews had been proposed many algorithms for multiple detection and tracking on the road without proper training mechanism rather than a threshold and off line tracking exists. In our proposed work, we described a new novelty method which is mainly based on the particle grouping method by using the two optimal filters namely Kalman and Particle filters based on the linearity motion. Kalman and Particle filters are chosen for preprocessing stages because it is known that the Kalman filter generates a best optimal estimate of the state of the object given in a set of measurements for a linear system while the particle filter proves to be successful in state estimation whereas the system is in dynamic and non-linear nature. Since any real-world system can never be completely linear or non-linear structure exists.

The main contribution of our work is presented as follows:

1) We proposed an architecture layer for object detection, estimation and selection and tracking are the main components by using Multi Layer Perceptron and Superposition Estimation for compute the potential and kinetic energy.

2) We developed the multiple tracking algorithm with the use of Kalman and Particle filter for tracking the location of a single and multiple object with the automation of the object detection engine.
3) We developed an occlusion state and its identification of occurrences of objects in a merged state with another.

The object detection phase involves detecting the target object from the sequence of frames that were grabbed from the given video after pre-processing, using the Haar training method and modified adaptive boost classifier approach that are applied to the system. The set of frames is then subjected to detection engine which returns a set of paired co-ordinates that depict the location of the object in the current frame of analysis. From these co-ordinates, the pixels are then grouped as particles which are fed into the linearity checker sub-system. In the linearity checker sub-system, the indices of the particle groups match with the next frame in sequence for tracking the motion of the object. The time intervals where the object is linear and non-linear are then identified so as to apply the Kalman and Particle filter for tracking the next state of the object in the video under analysis.

The rest of the paper is organized as follows. Section II discusses related works. Section III describes the proposed work and overviews are discussed in detail. The proposed algorithm and techniques is detailed in Section IV. Section V reports the experimental results. Section VI concludes this paper. Final Section VII discusses about the future work and its higher end applications.

II. Related Work
Several methods have discussed about the existing state-of-the-art of real time multiple object detection and tracking has sparked significant research in recent years. Although several studies have been performed, still it remains a highly challenging task due to abrupt changes in motion, camera and sensors, varying illumination and so on. Scientists and Researchers have more attention to solving the real time problems of traffic surveillance, accident avoidance, public safety and mortality rate reduction in every year is becoming increasingly more important so that we proposed a new techniques and algorithms in the field of computer vision and intelligent transportation sector to detect and track the multiple objects in the presence of occlusion. Still, there is a huge necessity to detect the target of multiple objects and track them effectively with occlusions and other included complexities exist multiple objects / vehicles in motion.

Bo-HaoChen et.al (2018) [7] has presented a sparse and low rank constraint model by using a contextual regularization approach for motion object detection. The benefit of this model yields more performance rate in a single scenario alone, hence there is a problem of finding multiple moving object detection in multiple scenario is challenging one. To overcome this problem our proposed method tested with the different data sets with occlusion and results are shown in the multiple scenarios of car race and multiple vehicles moving on the road.

Zhipeng Deng et.al (2018)[8] described the novelty of vehicle detection on region based convolutional neural networks(R-CNNs) in which combined a hierarchial feature maps to detect small object accurately. The combination of accurate vehicle proposed network with vehicle attribute learning network to found vehicles location and its attributes. Eventhough the vehicle detection mapped with the feature of objects there is a lack of training for the object detection engine and missed the linearity and non linearity motion on road. In our method we addressed the existing problems and shown in the experimental results using different datasets.

Key Lu et.al (2018) [9] has come up with the sparse window techniques to reduce the number of input image patches without accuracy. It causes a more drift in multiple vehicle detectors. Consequently parameter is too limited not fit for solving real time problems on/off road in multiple vehicles. In our proposed method solved the problems of multiple vehicle detection in the presence of occlusion and tested the performance of multiple objects/vehicles is shown in the results.

Yingfeng et.al (2017) [10] has used a scene adaptive vehicle detection algorithm based on a composite deep structure using Bagging method. The limitation found that only generation of target training samples with confidence scores so that it is not suitable for system automation to detect and track multiple vehicles on the road or dynamic environment. To overcome the existing issues, our
method satisfies with the multiple object tracking using haar trainer paired with adaptive discrete boost classifier and filters for linear and non-linear motion on the road.

Ravikumar Satzoda and Mohan Manubhai Trivedi (2016) [11] has proposed a vehicle detection using active learning and symmetry model in which detections are considered for rear view and eliminated the front view vehicles. To solve this problem we proposed a system automation for the object detection engine and features were collected and trained to the system to automation of multiple vehicle / object detection and track them effectively using multiple vehicle tracking algorithm and tested with the data sets and experimental results are shown by using MATLAB R 2016a.

Akshay et.al (2016) [12] came up with a bipartite matching problem to detect multiple hands online even in the presence of extraction of high level features. The key merit of this method was that it provided a comparison with generic MOT and delivered an exact solution for tracking multiple hands through bipartite matching technique. However, this is possible only through a series of assumptions made on background intensity, such as the static nature of the background, the illumination changes, the color of the objects, etc. In addition it was found to be viable only for self-occlusion. A better solution to this problem is proposed where in, the distance formulation technique based on bus topology is in use to detect the presence of partial and complete occlusions and training multiple models for multiple objects.

Dominick Kellner et.al (2016) [13] addressed the problem of integration of Doppler information of multiple detections from a single target for tracking and achieved over a large data set. The merit over this method was to detect multiple objects at single targeted objects in a frame sequence of 30 frames/Sec. The drawback noticed that absence of spatial assumptions on the rotation center of the rear axle. Our proposed work is to overcome the collisions and occlusions in multiple vehicle detection and tracking by particle grouping and fusion of optimal filters.

Harish Baskar et.al (2015) [14] has presented a method autonomous, multiple target detection and tracking for dynamic scenes. Thus the exact solutions for detection and tracking in spite of varying illumination changes, by Dynamic Reverse Analysis and Enhanced Rao-Blackwellized particle filter. However, this was achieved only through a series of assumptions is made in the background modeling with a limited number of available targets. In the proposed method, training is given for the object detection and hence no assumptions and limitations are made for object detection and tracking. Motion and appearances of moving objects in roundabouts, based on monocular system were stated by Hamid Hassannejad (2015) [15].

The method used to detect Haar-like features along with soft-cascade AdaBoost was used for detecting the moving vehicles. The detection rate yielded more accurate, but it limits the maximum speed to (i.e.) 6 km/Hr. The proposed method works on haar training algorithm and AdaBoost classifier for automatic detection of vehicles and there are no limitations on the speed.

A system for vehicle tracking and classification for data-driven approach was modeled by Sebastiano et.al (2015) [16]. The merit of this method was to monitor traffic flow up to 78.6% for moving objects. The drawback was limiting the application areas are occluded, collision were not handled by this method. The proposed work handles occlusions and avoids collisions in a dynamic environment.

Nazhao et.al (2015) [17] has worked on an adaptive partial occlusion segmentation method of locating the regions, optical flow and line scanning method for multiple vehicle tracking. The advantage of the method was that the candidate regions of occluded vehicles were presented effectively. This method failed in collision detection and collision avoidance. The proposed implementing the methodology concentrates on collision avoidance systems.

Chieh-Ching Wang et.al (2015) [18] has estimated Bus Topology and Interacting Multiple Model (IMM) using multiple hypothesis tracking algorithms are for tracking. This method covered limited area, i.e., urban area. The demerit of this method showed that consumed very high power when
The proposed work tracks multiple vehicles by machine learning detection. So that the accuracy will be attain more.

The feature fusion technique for robust object tracking using fragmented particles in [19] uses Particle filter to track a single object within a single camera and a Blob-based target association scheme for tracking across cameras. The advantages of this work are that it withstands illumination changes and detects partial occlusions. The disadvantage is that it fails to work in dynamic background and also in detection of same colored objects. In the proposed work dynamic scene modelling along with motion based energy parameters are effective in solving the static background problem.

Robust human tracking algorithm applied for occlusion handling in [20], was proposed for single target tracking with color and motion integration mechanism using particle filters. The advantages of the system were that single object was tracked even in cluttered background and also partial and full occlusions were handled. But the disadvantage of the approach was that overlapping objects go undetected even in a static background. The solution to overcome this detection problem is the automation of the object detection engine along with the proposed particle grouping approach.

It can be summarized from the existing literature that the particle filter proves to be versatile in the field of application of object tracking as it addresses the multiple object detection in varying motion with occlusions.

III. Overview of the Proposed Method

A. Single Object Tracking

The system architecture is a multi-layered framework which involves the fusion of filters for single object tracking as depicted in Figure 1.

In general, object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. But a serious flaw of this approach is that the background cannot be modeled in a generic manner and added to that, changing backgrounds would lead to an increase in the false positive rate. Thus, there is a necessity for a generic and the optimal object detection engine to meet the requirements of an efficient tracking system.

A simple rectangular Haar-like feature considers adjacent rectangular regions, at specific locations where objects are from the frame, sums up the pixel intensities of those regions and calculate the difference between the rectangular regions. In the detection phase of the object tracking framework, a large set of positive and negative images is collected. A positive image is the one that has the desired target depicted in it while a negative image is one that does not have the target. As a simple example for car tracking, an image of a car is said to be positive image while an image of an empty race track or even any other vehicle is considered to be a negative image.
After image acquisition, the object marking is necessary. The object marking is done for the set of positive images where each object in each image is marked by a rectangle and the corresponding coordinates of the rectangle are stored in a text file. These are then utilized for identifying features of the target and they are stored in a tree like structure in an XML file.

For each subsection of the input frame the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. The object detected is then visually represented by means of a rectangle with the length and width corresponding to the matched pixel coordinate values. Haar-like feature is a weak learning algorithm with minimum classification error and it is paired with adaboost classifier, which is an iterative learning algorithm to construct a strong classifier. It rectifies the misclassification of subimages done by haar.

After detecting the object, only the rectangular coordinates are considered instead of the entire frame and further processing is done only for the region of interest. The pixels in the region of interest are grouped sequentially to form rectangles as shown in Fig.2b. Particle grouping is done based on the dimensions of the object in the image.

After the object detection phase the co-ordinates of the object are known and thus grouping is done based on the height and width of the object that is detected. Those particles that do not fit exactly into a rectangular group are considered as a single group of particles which are then weighed during the re-sampling phase.
After particle grouping, the indices are alone matched with the indices of particle groups generated for the incoming frames. When a match occurs the corresponding group indices are stored and then the position of the object is plotted on a graph against time.

**Figure 2:** Detected object (car)

![Figure 2](image)

In Figure 2 (a), the red car is detected and depicted in a bounding rectangular box. The pixels in the rectangular region are divided into groups of particles based on the height and width of the object. Here, the 520 x 480 image is equal divided into groups, each of a size such that a major portion of the car is bounded within the rectangle. This is represented in the following Fig. 2 (b) as rectangles from R1 to R21.

It can be inferred from the Figure. 2 (b) that the groups of particles in R1 and R5 do not form a part of the car and that the rectangular group R21 has parts of it which do not form the part of the car. However, the car cannot be exactly bounded in a rectangular box because of its transitional motion and hence the remaining particles are grouped under a single rectangle (R21). These particles may be discarded during the re-sampling process of particle filtering such that they are filtered as measurement noise.

The Fig. 2 (c) indicates that the groups R1 and R5 are discarded (marked visually with a cross mark over the rectangle) as they are a part of the background and are not considered while processing the frame. Only the groups R2, R3, R4, R6 upto R21 is matched with the next frame in sequence. In this way, the number of particle matching can be reduced by 50%.

Tracking the object’s true position is done by tracking its state. This uses information from the object’s features and the previous object state to create an estimate of the object's new state. The combination of previously estimated object dynamics and current measurements helps eliminate noise from measurements that would otherwise lead to erratic object tracking. As a simple example, knowing the previous position and velocity of a car allows us to give a rough estimate of its current position. When combining this estimate with additional information of its state, tracking accuracy can be overall improved.

In the proposed system, the object tracking component comprises of two filters, namely Kalman and Particle filter that are used appropriately for processing the objects in the video under analysis. The Kalman Filter addresses the basic problem for estimation of the state in a discrete-time controlled process [21]. Though Kalman filters provide a convenient measure of estimation accuracy and fuses information from multiple-sensors, the Kalman filter can’t be applied to nonlinear systems. So in such cases, the Particle filter is applied.

Particle filter is based on point mass representations of probability densities, which apply to any state model [22]. Particle Filter is a hypothesis tracker, which approximates the filtered posterior distribution by a set of weighted particles [23]. It weights particles based on a likelihood score and then propagates these particles according to a motion model.

Thus the object in the frame, where its motion is non-linear, is tracked by estimating the motion parameters of the particles and the state parameters of the object. Finally the object is tracked in a continuous stream of frames by combining both the output of Kalman and Particle filter.
B. Multiple Object Tracking

System Architecture
Input Layer
A video is read at the input which is then split into a set of frames at the rate of 30 frames per second. These video sequences are given as input to the object detection layer.

Object Detection Layer
The supervised learning approach is used in order to train the system with the target object for automatic detection. The Haar training approach is used which is followed by a strong classification method using the Adaptive boost classifier. The result of this engine is the coordinate of the bounding rectangle box that encloses the object.

In the fig. 3. Discusses about the layers of particle grouping, each group is associated with a descriptor value and the strength of identity and it is computed based on the visual cues of the object. The maximum identity, strength group is used for identifying the whole object uniquely. Further processing of the object is done only using this identified particle group. Edge detection for the pixels in the identified particle groups is done and then the array values are binary coded and stored. Only the binary values are stored because in case the object moves out of frame and re-appears a few seconds later, it is identified correctly and matched using this binary value. This proves to be useful whenever an existing object reappears in the scene.

This identity recognition is done initially by identifying the objects in the scene. As an example, in a car race, all the cars participating in the race are identified uniquely by corresponding particle groups. Since the cars and the number of cars are known at the start itself, the identity recognition is done at the start and no later.

Using the coordinates of the particle groups identified the bus along with the links to the objects in a scene is created. The line connecting the object to the bus is referred to as the link. This varies from frame to frame depending upon the position and presence of the object under motion.

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. An object tracker may also provide the complete region in the image that is occupied by the object at every time instant.

The tasks of detecting the object and establishing a correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm, and then the tracker corresponds to objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames.

A simple rectangular Haar-like feature considers adjacent rectangular regions, at specific locations where objects are from the frame, sums up the pixel intensities of those regions and calculate the difference between the rectangular regions. In the detection phase of the object tracking framework, a large set of positive and negative images is collected. A positive image is the one that has the desired target depicted in it while a negative image is one that does not have the target.
Tracking the object’s true position is done by following its states. This uses information from the object’s features and the previous object state to create an estimate of the object’s new state. The combination of previously estimated object dynamics and current measurements helps eliminate noise.
from measurements that would otherwise lead to erratic object tracking. As a simple example, knowing the previous position and velocity of a car allows us to give a rough estimate of its current position. When combining this estimate with additional information of its state, tracking accuracy can be overall improved.

Though its non-linear counterparts EKF and UKF exist, they are not optimal estimator. In addition, if the initial estimate of the state is incorrect or if the process is modelled incorrect, the filter may quickly diverge, owing to its linearization in fig. 4. In case of Particle filter, a major issue lies in the complexity of the particle set that is the number of particles required to accurately represent the object’s motion. The number of particles to be chosen for each object is exponential in nature due to its highly varying motion and divergence between the predicted and target (object) states. This pays a heavy penalty in real time, tracking scenarios.

To overcome these serious problems, the proposed system performs particle grouping based tracking with multi-layer perceptron learning technique for weight computation and assertion. This reduces the number of computations by 40% as the process of tracking is done using the identified particle groups alone. Also the system proves to work well under linear and non-linear motion.

**Occlusion Handling**

Occurrence of occlusion is a major part in multiple object tracking and the proposed work is addressed the problem of multiple moving vehicles / objects in motion is shown in Fig. 5. with their presence of occlusions.

**Figure 5: Occlusion Detection and Handling**

- **Step 1:** Identifying the Occurrence of Occlusion
- **Step 2:** Tracking when Objects are in Entangled State (merge)

**IV. Proposed Algorithms and Techniques**

**Algorithm:** Multiple Objects Tracking Algorithm

**Input:** Video (V) realized into a set of frames (Fr)

**Output:** Location points \((x^*, y^*)\) of the objects in video where \(O_i\) represents Object \(i\) at time ‘t’ \(X_{i,t}\) represents state of the object \(i\) at time ‘t’ and \(Y_{i,t}\) state and \(W_{i,t}\) weight association property, \(K_{i,t}\) Kinetic energy and \(P_{i,t}\) potential energy.

**Step 1:** Particle set generation

For Object \(O_i\), Particle set can be defined as in Eq. 1.
\[ X_{i,t} = \left\{ (x_{i,t}, y_{i,t}, w_{i,t}^{(s)}, K_{i,t}, P_{i,t}) \right\}_{s=1}^{N_s} \]  

The \( X_{i,t} \) value is the state of the object \( i \) at time \( t \) which is represented as four parameters namely the state and weight association property, Kinetic energy and potential energy.

Kinetic Energy

The Eq.2 is the kinetic energy of the object \( i \) at time \( t \). It is a function in mass \( m \) and the velocity \( v \). The kinetic energy of a body exists only when it is in motion as signified by the velocity estimate.

\[ K_{i,t} = \frac{1}{2} m_i v_{i,t}^2 \]  

Case (i) object from stationary to start state

When an object is initially stationary or at rest, then \( v = 0 \). Therefore no kinetic energy but potential energy exists. When the object suddenly starts, \( v = v_o \).

\[ K_{i,t} = \frac{m v_o^2}{2} \]  

Case (ii) Object in continuous motion

When an object is in continuous motion with a linear increase in speed from time to time, for a given time \( t \) and velocity \( v_{i,t} \)

\[ K_{i,t} = m v_{i,t}^2 \text{ With } \frac{dv}{dt} = \text{constant} \]  

Case (iii) Sudden acceleration

When there is a sudden acceleration in motion then, \( \frac{dv}{dt} \neq \text{constant} \). For an example, when a car suddenly accelerates to race its counterparts, say during a lap start state or lap finish state, kinetic energy increases abruptly. Then

\[ K_{\text{inc}} = \frac{m v_{\text{inc}}^2}{t} \text{ since } \frac{dv}{dt} \neq \text{constant} \]  

Case (iv) Sudden deceleration

When the sudden deceleration occurs, then there is a drop in velocity such that \( v = v_{\text{dec}} \). This implies that the kinetic energy decreases, may or may not reach a minimum value.

\[ K.E = K_{i,t}^{\text{dec}} \quad K_{\text{dec}} = \frac{m v_{\text{dec}}^2}{t} \]  

For example in a car race, sudden deceleration occurs when it takes a steep turn in the track or crosses hurdles/speed breakers.

Case (v)

When the object comes to a sudden halt \( v = 0, K_{i,t} = 0 \)

Potential energy:

The Eq.7 represents the potential energy of the object \( i \) at time \( t \). It is a function of mass \( m \), gravity \( g \) and the distance \( d \). Though the object is under motion \( m \), the \( d \) parameters are not subjected to change while the distance between the center of mass and the track in which object moves varies from time to time.

\[ P_{i,t} = m_i \cdot g \cdot d_{i,j} \]  

The Eq. 8 gives the distance between object \( i \) and object \( j \) at time \( t \) and \( t+1 \). This distance value decides the important parameter in the computation of the potential energy.

\[ d_{i,j} = \frac{\sqrt{\left( (x_i^t-x_{i+1}^t)^2 + (y_i^t-y_{i+1}^t)^2 \right)}}{\max |d|} \]  

Step 2: Aggregate particles into groups

The Eq 9 represents the particle groups of object \( i \) at time \( t \) which is a collection of groups \( g_{l,m} \) where \( l \) and \( m \) parameters depict the frame width and height respectively.

\[ G_{i,t} = \{ g_{l,m} | l < Fr_{t,w}, m < Fr_{t,h} \} \]  

Step 3: Identify the identity particle groups,
The main goal of the identity recognition is to provide a measure of importance to a few particle groups with assertions of detecting the object \( O_i \)

### I. Identity Function

Let \( D = \{ A_0, A_1, A_2, \ldots, A_n \} \) be the individual descriptors, \( d^i \) be the possible subset of assertions in \( D \), \( s^i \) be the strength of the identity, then Eq. 10 is obtained where \( d^i \) maps to \( s^i \). \( P(D) \) denotes the power set of \( D \), \( N_d \) the number of different assertion values and \( FP \) is the number of feature properties. A challenge during this is that same individual object should not be described in many ways.

3.1 Identify the visual prompts of the object \( O_i \)

(i) \( I(D) \rightarrow \{ (d^i, s^i) \}, \forall d^i \in P(D) \)

(ii) \( d^i \) maps to \( s^i \) and

(iii) Compute

\[
\forall d^i \in P(D), d^i \rightarrow \frac{N_d}{FP} \quad (10)
\]

### II. Matching Function

Goal: Provide a measure of similarity between \( D^1 \) and \( D^2 \)

In Eq. 11, \( S \) signifies the measure of strength of similarity between \( (D^1 & D^2) \) and \( D^1, 2 \). This implies that higher the descriptors’ similarity then the higher is the probability that \( D^1 \) & \( D^2 \) denote the same real object.

3.2 Mapping function

(i) Identify \( D^1_O = \{ A^1_0, A^1_1, A^1_2, \ldots, A^1_n \} \) descriptors for object 1

\( D^2_O = \{ A^2_0, A^2_1, A^2_2, \ldots, A^2_n \} \) descriptors for object 2

(ii) Compute \( I(D^1, D^2) \rightarrow D^{1,2}, s \)

\[
(11)
\]

3.3 Bus Topology Creation

(i) \( \forall t, j \sum_{i:j \leq N(i)} F_{ij}^{t-1} = M_j^t = \sum_{k=N_j} F_{ij,k}^t \quad (12) \)

(ii) \( c(n_{ij,k}^t) = -\log \left( \frac{\rho^t_{ij}}{1 - \rho^t_{ij}} \right) \quad (13) \)

(iii) Calculate the co-ordinates of bus on the track from the particle group

\[
T_{ij,k} = \sum_{i,j,k} X_{n/2,j} \cdot Y_{n/2,j} \quad (14)
\]

(iv) Co-relate the bus with estimated number of objects

\[
N_{ij,k} = \{ T_{ij,k}, M_j^t \} \quad (15)
\]

Step 4: Prediction of Object States

For particles in the identified group alone, predict the state as in Eq. 16

\[
X_{t,t-1} \rightarrow X_{t,t}^{pred} \left( \{ x_{i,t}^{s(s), w_{i,t}(s), K_{i,t}, P_{i,t}^t} \}_{s=1}^{N_s} \right) \quad (16)
\]

\[
x_{i,t}^{s(s)} = F(x_{i,t-1}^{s(s)}, w_{i,t-1}(s)) \quad (17)
\]

\[
\Delta w = \eta . x_{i,t}^{s(s)} \cdot y \quad (18)
\]

\[
w_{i,t}(new) = w_{i,t}(old) + \Delta w \quad (19)
\]

Where the x and y component are represented as in Eq. 17 and Eq. 18. And \( F \) is the state transition model that consists of 2 parts:

1. Deterministic parts
   - Weight Association
   - State of the object
2. Stochastic parts
   - Random visual dependencies of the object
Random acceleration of the objects and hence Kinetic energy becomes random

Step 5: Calculating Likelihood score for the identifying particle groups

For the identifying particle groups, \(G\left(x_{i,t}^{*}\right)\), likelihood function is computed as in Eq.20. It indicates whether an object touches or matches with the feature \(m\) or not. It implies the difference between the observed and estimated states.

\[
LS = l\left(m, x_{i,t}^{*}\right), l\left(m, x_{i,t}^{*}\right) \quad (20)
\]

\[
LS\left(Z_t \mid x_{i,t}^{*}\right) = l\left(m, x_{i,t}^{*}\right), l'\left(m, x_{i,t}^{*}\right) \quad (21)
\]

For the particle group determine,

1) \(G\left(x_{i,t}^{*}\right)\) => groups that object identity lies in
2) \(l(.)\) => match set
3) \(l'(.)\) => non-match set

Step 6: Updating occlusion estimates

(6.1) Superposition estimation computation

\[
\bar{E} = \sum_{i=1}^{N} K_{i,t} + P_{i,t} \quad (22)
\]

Updating the occlusion estimates, can be done by computing the energy values of the object as represented in Eq.22 and Eq.23. If \(\frac{\partial \bar{E}}{\partial t} = 0\), then occlusion has occurred and hence the merge state has to be resolved. Since the object is under motion, its states are time varying and time dependent. Thus the energy is computed in the time dependent equation Eq. 23

(6.2) Compute time variant energy values

\[
H = i. h. \frac{\partial \bar{E}}{\partial t} \quad (23)
\]

(6.3) If \(\frac{\partial \bar{E}}{\partial t} = 0\), then occlusion has occurred and merge state has to be resolved.

Step 7: Distance Formulation technique

(7.1) Calculate the prior distance of the object states to track

\[
Z_{i,j,k}^{prior} = \sum_{i,j,k} \left[ (D_{p_{i,k-1}} - D_{p'_{i,k}})^2 + (D_{q_{i,k-1}} - D_{q'_{i,k}})^2 + (D_{r_{i,k-1}} - D_{r'_{i,k}})^2 \right] + nc_{i,j,k-1} \quad (24)
\]

(7.2) Calculate the posterior distance of the object states to track

\[
Z_{i,j,k}^{post} = \sum_{i,j,k+1} \left( (D_{p_{i,k-1}} - D_{p'_{i,k}})^2 + (D_{q_{i,k+1}} - D_{q'_{i,k}})^2 + (D_{r_{i,k+1}} - D_{r'_{i,k}})^2 \right) + nc_{i,j,k+1} \quad (25)
\]

Step 8: Entanglement free treatment

(8.1) Calculate the prior distribution of object states

The prior distribution of object states is given by Eq.26. The prior particle set \(X_{i,t}^{prior}\) representing the prior state distribution for updated hypotheses is obtained as in the two equations Eq.26 and Eq.27. The weight of the prior set is proportional to the product of the prior weight set and importance weights of the identity group.

\[
X_{i,t}^{prior} \rightarrow X_{i,t}^{prior} = \left\{ \left( x_{i,t}^{*}, w_{i,t}^{prior}, E_{i,t}^{prior} \right) \right\}_{s=1}^{Ns} \quad (26)
\]

\[
w_{i,t}^{prior} \propto w_{i,t-1}^{prior} \cdot \sum w_{i,t}^{prior}(G) \quad (27)
\]

(8.2) Calculate the posterior distribution of object states
The posterior distribution of the object states is given by Eq.28. The weight of the posterior set is proportional to the product of the likelihood score of the observed and the measured states and the posterior weight set of the identity group G.

\[ X_{i,t}^{\text{prior}} \rightarrow X_{i,t}^{\text{post}} = \left\{ \left( x_{i,t}^{s'}, w_{i,t}^{\text{post}(s)}, E_{i,t}^{\text{post}} \right) \right\}_{s=1}^{N_g} \]  

\[ w_{i,t}^{\text{post}(s)} \propto \text{LS} \left( z_t \right) x_{i,t}^{s'} \sum w_{i,t}^{\text{post}(s)} \left( G \right) \]  

(8.3) Compute Covariance

The covariance is the measure of deviation of the observed and estimated mean values. The main aim is to make the covariance value minimum, if not negligible. If covariance is minimum or negligible, then

- Energy is nullified due to overlap
- Occlusion has occurred
- Mark identifiers and continue tracking
- If large covariance value is reckoned then
- No occlusion has occurred or object is in split state
- Re-compute weights and perform state prediction.

\[ \text{Cov} = E \left[ (X^{\text{pred}} - E[X]^{\text{pred}})(X^{\text{post}} - E[X]^{\text{post}}) \right] \]  

\[ E(X) = \frac{x_1 p_1 + x_2 p_2 + \cdots + x_k p_k}{p_1 + p_2 + \cdots + p_k} \]  

P for i=1 to k are particle weights

E (X) value gives the weighted average.

The Eq.31 represents the expectation value.

Step 9: Go to step 4 until next frame exists.

[End of algorithm]

Here, we briefly discusses the theorems and list out how to solve the occlusions between objects in real time motion.

Theorem1: Occlusion between objects can be detected and handled by the bus topology created for each frame

Proof: Consider two objects A and B which are connected to the bus as shown in Figure 6. Let \( C_a \) and \( C_b \) be the centroids of A, B where ‘a’ and ‘b’ are the distance between the centroid and the bounding box of the objects A, B respectively. Also d is the distance between the centroids of the objects and \( d_{ab} \) is the distance between the bounding boxes between the objects A and B. Then

\[ d_{ab} = d - a - b \]

With this distance value from subsequent frames the object’s motion can be relatively identified. Using this parameter, the correspondence of the objects can be detected. When the corresponding matches then it indicates the occurrence of occlusion. Thus the approach emphasizes and actuates occlusion handling between objects.

Lemma 1 (No occlusion):

If A and B are at a distance \( d_{ab} \) apart at the \( i^{th} \) moment, \( d_{ab} \) apart at \( j^{th} \) moment or \( d_{ab} \) at \( k^{th} \) moment as shown in Figure 10 a,b,c then it is said to be that the objects A and B are not occluded.

Proof:

i. \( d_{ab} = d - a - b \)

ii. \( d_{ab} = d - a - b \)

In both cases, \( d_{ab} \) and \( d_{ab} \) are greater than zero that is objects A and B are moving with a small distance apart from each other and are still completely and individually visible.
Figure 6: Objects A and B

If A and B are moving in such a way that they are one behind the other as shown in Figure 8, then $d_{ab}=0$. Also the set of indices in the segment $l_a$ does not match with that of $l_b$ indicating that neither of the objects or even part of the objects occlude each other. It simply signifies that the objects are in close proximity.

Figure 7: Objects A and B when in Close Proximity

Lemma 2 (Partial Occlusion):
If the objects A and B are moving in such a way that they appear as in Figure 8.a, then occlusion is partially detected.
Proof:
If the distance between the objects
Distance $d_{ab} < 0$ then part of object B is occluded with A.
Figure 8: Occlusion Handling Specifications:

- a. Start of partial occlusion (merge state)
- b. B overtakes A (occlusion -merge state)
- c. Occlusion is over (split state)

A, B – objects, C – center, d – distance

But it is notable that the object A may race ahead of B rendering the value $d_{ab} < d_{ba}$ that is $d_{ab}$ will be more negative when compared to $d_{ba}$ indicating that the objects travel/move under occlusion (entangled state). In such a state, the index of C_b will be lower than that of C_a and hence $d_{ab}$ is computed as $d_{ba}$ representing the split state as in Figure 8 b, 8c.

Lemma 3 (Complete Occlusion between objects):

When both the objects A and B are moving as shown in Figure 9, then the objects are totally occluded.

Proof:

Upon comparing the set of indices on segment $I_a$ and $I_b$ are the same.

$d_{ab} = d-a-b$

Here d=0 and a=b (since the object is completely occluded),

Then $d_{ab} = -2a$ or -2b

Figure 9: Objects A and B when in the Complete Entangled State
V. Performance Analysis
A. Single Object Tracking

In this section, we evaluate our method for single and multiple object tracking. Experiments were implemented based on the filtering techniques and executed on a system using MATLAB 16.0.0.a.

Two datasets were used in the experiments. The first dataset was a publicly available vehicle dataset, which was collected over the city of Munich Car race on road. To provide further verification, we also evaluated our vehicle tracker and detector on another collected vehicle dataset, which contains 50 images in each datasets.

The training and testing were evaluated in the input video with the presence of multiple moving objects like car, bus and lorry. The video is split into a number of frames using the frame splitter algorithm and the frames are stored as images. The video is split into a sequence of frames at the rate of 4 frames per second.

Each frame is saved in the JPG format and the frames are fetched during the execution of her training for system learning. From the positive set of images, the object in each frame is marked manually so as to help the system learn about the target in the image. The corresponding coordinates of the rectangular marking object are stored in a text file along with the name and location of the image, the number of objects in each image.

Object Detector

The set of frames from the input video is fed into the object detector subsystem at the rate of 30 frames per second. In each frame, pixel by pixel matching is done with the set of features of the target that is stored in the XML file after the execution of the system trainer.

The object detector is executed to detect the performance of training and the yielded output is as shown above in Figure 10. The number of false positives was found to be minimized with an increase of 30% in performance after system training.

Figure 10: Object Detected in Frame 1

It can be inferred from the Figure 11.(a) that the groups of particles in R1 and R5 do not form a part of the car and that the rectangular group R21 has parts of it which do not form the part of the car. However, the car cannot be exactly bounded in a rectangular box because of its transitional motion and hence the remaining particles are grouped under a single rectangle (R21). These particles may be discarded during the re-sampling process of particle filtering such that they are filtered as measurement noise.
**Figure 11:** Object Detection by haar training and Adaboost classifier

![Object Detection](image)

**Table 1:** Space Analysis for Different Sized Object based on Particle Grouping

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frame size</th>
<th>Car size</th>
<th>Groups divided</th>
<th>Groups dimension</th>
<th>Matrix size</th>
<th>Total groups</th>
<th>Remaining particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>420x240</td>
<td>282x92</td>
<td>20</td>
<td>50x21</td>
<td>5x4</td>
<td>250x84</td>
<td>4944</td>
</tr>
<tr>
<td>2</td>
<td>130x480</td>
<td>130x105</td>
<td>20</td>
<td>25x25</td>
<td>5x4</td>
<td>125x100</td>
<td>1150</td>
</tr>
<tr>
<td>3</td>
<td>640x480</td>
<td>120x55</td>
<td>18</td>
<td>20x15</td>
<td>3x6</td>
<td>120x45</td>
<td>1200</td>
</tr>
</tbody>
</table>

The Haar training method with the Adaboost algorithm for automation of the object detection engine exhibits high accuracy.

**Table 2:** Accuracy of System Learning

<table>
<thead>
<tr>
<th>No.of.Images</th>
<th>Accuracy</th>
<th>False Positive Rate</th>
<th>False Negative Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>85.9%</td>
<td>25.7%</td>
<td>31.6%</td>
</tr>
</tbody>
</table>

Checking For Linearity In Motion Of The Object

Once grouping of particles of the object has been done, the linearity of the object’s motion is checked by matching the particle group indices with the current frame of the video under analysis and the corresponding co-ordinates that are matched are stored in an array.

**Object Tracker**

In the data set 1, Figure 12.a) the car is in linear motion until it takes a turn near the curve point of the track. During the turn it can be noted visibly that the car’s motion is completely non-linear. The identifying particle groups are successively matched to the car in the next frame and in case it is found...
missing in the successive frame the neighboring particle groups are matched with to confirm the presence or absence of the car.

**Figure 12: Results of the Object Tracker for Data Set 1**

![Figure 12](image)

The object’s position in the environment is plotted as points against the time period (Figure 12.b). The points are obtained as a result of the object detector subsystem. The particle group index refers to the index of the frame that matches with the index of the particle group of the object in the frame.

In the data set 2, (Figure 13.a) the car moves in a zigzag manner such that there is continuously switching between linear and non-linear motion. The graph (Figure 13.b) shows the movement of the object across a sequential time interval. The points of interest are the steep curved co-ordinates which indicate the drastic change in speed of motion of the object. Such points are indicators of the non-linearity in the object’s motion.

**Figure 13: Results of the Object Tracker for Data Set 2**

![Figure 13](image)

In the data set 3, (Figure 22.a) the car moves in a linear manner such that there is path traversed by it is along a straight track. This is portrayed through the graph as in Figure 22.b.

**Figure 14: Results of Object Tracker for Data Set 3**

![Figure 14](image)
In a frame of dimensions 640x480, the car’s size is 120x55 pixels. Based on this size, the car is divided into 18 groups each of 20x15 dimensions. The groups together form a 3x6 matrix with a total of 120x45 pixels. The remaining 1200 particles are utilized or discarded in the process of tracking.

**Comparison Analysis of Proposed System for Single Object Tracking**

It is inferred from the results that the system which presents a randomized approach for object tracking with a grouping of particles for motion based object tracking using a combination of Kalman and Particle filter enhances system’s performance by 30%.

The system was tested with videos having moving objects in varying backgrounds and along different trajectories. Firstly, the object was tracked using Kalman Filter by estimating the state of the object which was considered as a single unit. (Figure 15.a) Secondly the object was represented as a collection of particles and tracked using Particle filter alone and finally the object was sub-divided into particle groups (Figure 15.b) and tracked by combining Kalman and Particle filter.

**Figure 15:** Response Time Comparison

![Graph showing response time comparison between Kalman Filter and Particle Filter](image)

a) Kalman Vs Particle Filters  
b) Frame Number VsParticles

When the system was subjected to a video of length 11 seconds, where the object traverses partly in linear motion and partly in non-linear motion as depicted in Figure 15.b, it was found that the response time of the particle filter was reduced by nearly 50% in comparison with that of the Kalman Filter.

**Figure 16:** Response Time Comparison between Particle Filter and Particle Filter after Particle Grouping

![Graph showing response time comparison for different filters](image)

a. Response Time Comparison of Filters for Data Set 1  b. Response Time Comparison of Filters for Data Set 2

The graph (Figure 16.a) shows the variation in the execution time of the system when rendered in the aforementioned scenario. For a frame of resolution 350x240, a total of 2040 particles was
required. But in the proposed system, due to grouping of particles it is reduced to 1050 particles which is approximately 50% reduction in computation.

The graph (Figure 16.b) shows the response time comparison of Kalman filter, Particle Filter and particle grouping for data set 1. The nature of data set 1 (Figure 16.a) is that the object is linear in motion until it takes a turn and moves in a non-linear motion as depicted in Figure 16.b. In this scenario, the response time drops from 4s to 1s in case of Kalman and the proposed approach while it is halved in case of Particle and the proposed system.

The graph (Figure 16) shows the response time comparison of Particle Filter and combination of Kalman and Particle based on particle grouping of data set 2.

The nature of data set 2 (Figure 16.a) is that the object moves in a complete zigzag path with frequent switches between linear and non-linear motion (Figure 16.b). Since the track traversed by the car is fraught with turns, there is frequent switching between the Kalman and Particle filter for processing the state of the object. But in this environment, the mere Kalman filter alone will not work as it fails to track objects in non-linear motion. Hence in Figure 16, comparison is made between Particle filter without grouping and with grouping.

It can be inferred that though there are more switches between the two filters, the average response time of particle filter without grouping is 2.93s while that after grouping is found to be 2.88s. Still, the system improves the throughput by 5% in comparison with the existing approach.

The graph (Figure 17) shows the response time comparison of Kalman filter, Particle Filter and combination of Kalman and Particle based on particle grouping of data set 3.

**Figure 17: Response Time Comparison of Filters for Data Set 3**

![Response Time Comparison Graph](image)

The nature of data set 2 (Figure 17) is that the object traverses along a completely linear path (Fig.16.b). It is found that even in case of complete linear path, the system provides a lesser response time in comparison with the estimate of Kalman filter because of particle grouping. The average response time in case of the Kalman filter is 2.25s, Particle filter is 2.37s but on combining it is 1.97s. The accuracy of the proposed model for single object tracking for three different data sets on an average was found to be 85% but the missing detection rate were 26% on an average.

**Table 3:** Accuracy for Single Object Tracking

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Video resolution</th>
<th>Object Size</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>420x240</td>
<td>282x92</td>
<td>9~11 fps</td>
</tr>
<tr>
<td>II</td>
<td>640x480</td>
<td>130x105</td>
<td>13~15 fps</td>
</tr>
<tr>
<td>III</td>
<td>640x480</td>
<td>120x55</td>
<td>12~15 fps</td>
</tr>
</tbody>
</table>
Table 4: Speed Analysis for Single Object Tracking

<table>
<thead>
<tr>
<th>Nature of the Dataset</th>
<th>Accuracy</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>L&amp;NL</td>
<td>81.7%</td>
<td>28.7%</td>
</tr>
<tr>
<td>Highly NL</td>
<td>84.1%</td>
<td>26.8%</td>
</tr>
<tr>
<td>Highly L</td>
<td>89.5%</td>
<td>22.5%</td>
</tr>
</tbody>
</table>

SO – Single Object, L- Linear, NL- Non-linear

The speed of the proposed model was analyzed by subjecting the system to different datasets where the object size was variable in nature. On an average it was found that the speed of the tracking (Table 4) in case of single object was found to be as high as 13 fps. This is approximately 46% increase in speed in comparison to the approach proposed in [20].

A. Multiple Object Tracking

From the stored set of frames, separately classify the set of positive images (images that have the target object present in them) and the set of negative images (images that do not have the target object but have the environment present in them).

Table 5: Space Analysis for Different Sized Object based on Particle Grouping

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frame Size</th>
<th>Car Size</th>
<th>Groups divided</th>
<th>Groups Dimension</th>
<th>Matrix Size</th>
<th>Total groups</th>
<th>Remaining Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>480x360</td>
<td>142x60</td>
<td>12</td>
<td>30x20</td>
<td>4x3</td>
<td>120x60</td>
<td>1320</td>
</tr>
<tr>
<td>5</td>
<td>640x479</td>
<td>230x180</td>
<td>30</td>
<td>40x30</td>
<td>5x6</td>
<td>200x180</td>
<td>5400</td>
</tr>
<tr>
<td>6</td>
<td>640x332</td>
<td>290x110</td>
<td>28</td>
<td>40x25</td>
<td>7x4</td>
<td>280x100</td>
<td>3900</td>
</tr>
</tbody>
</table>

Figure 18: Trainer Output

From the positive set of images, the object in each frame is marked manually so as to help the system learn about the target in the image. The corresponding coordinates of the rectangular marking object are stored in a text file along with the name and location of the image, the number of objects in each image.
Object Detector

The set of frames from the input video is fed into the object detector subsystem at the rate of 20 frames per second. In each frame, pixel by pixel matching is done with the set of features of the target that is stored in the XML file after the execution of the system trainer.

Figure 19: Detected Object

The object detector is executed to detect the performance of training and the yielded output is as shown above in Figure 30. The number of false positives was found to be minimized with an increase of 30% in performance after system training. Different data sets were collected to analyze the working of the proposed system under varying object motion. Each car in motion is tracked using a bounding rectangular box using a different color for each car.

Figure 20: Results of the Proposed System for Data Set 4

In the data set 4 (Figure 20), the cars travel along a linear road with steep turns in laps between. The Figure 20 depicts that in spite of sudden turn points in the track, the proposed system effectively tracks all the cars in motion. Also there are very minimal or negligible false positives (Table 5) in this data set. The graph (20 b) shows the response time comparison of Particle Filter and proposed approach based on particle grouping of data set 4. The nature of data set 4 (Figure 20 a) is that the objects travel in linear motion until they take a turn and move in non-linear motion. In this scenario, the response time has dropped by 0.5s in comparison with the 3.5s of Particle filter.
The nature of data set 5 (Figure 21 a) is that the objects move in a complete chaotic manner through a race track. In the data set 5 (Figure 21), the cars travel along a straight road but the motion of the cars is completely chaotic. Though the clarity of the cars in the video is poor due to illumination problems, the proposed system tracks the two cars in scene accurately. The number of false positives in this data set is acceptable if not negligible. The graph (Figure 21 b) shows the response time comparison of Particle Filter and the proposed system for data set 5.

**Figure 22: Results of the Object Tracker for Data Set 6**

In data set 6 (Figure 22), the two cars travel over a bridge racing each other. The trajectory of the race is zig-zag in manner. The proposed object tracker tracks the two distinct cars that are linear in motion correctly and efficiently. The graph (Fig 22 b) shows the response time comparison of Particle Filter and that of the proposed system for data set 6. The nature of data set 6 (Figure 22 a) is that the objects traverse along a zig-zag bridge racing in a linear as well as non-linear motion. It is found that even in case of background obstruction in the visibility of the moving objects, the system provides a lesser response time in comparison with that of the existing approach. The average response time in case of Particle filter is 4s but after particle grouping it is 3.25s which are approximately 3% lesser.
Performance Analysis of the Proposed System for Multiple Object Tracking

It is inferred from the results that the proposed system which presents a particle grouping based approach with bus topology based distance formulation technique for object tracking enhances system’s performance by 40% in comparison to the object tracking system using a combination of Kalman and Particle filter which increased only 30%. The system was tested with videos having moving objects in varying backgrounds and along different trajectories. Firstly, the object was represented as a collection of particles and tracked using Particle filter alone and secondly the object was sub-divided into particle groups and tracked by using the proposed system.

When the system was subjected to data set 5 the object traverses partly in linear motion and partly in non-linear motion, it was found that the response time of the proposed system was reduced by nearly 50% in comparison with that of Particle Filter. It can be inferred that though there is a random motion of the objects, the average response time of particle filter without grouping is 4.5s while that after grouping is found to be 2.96s. Thus, in this scenario where the video and the illumination clarity lack high-end quality, the proposed system improves the throughput by 35% in comparison with the existing approach.

Occlusion Detection

Partial Occlusion

The accuracy in tracking the moving objects under various types of occlusion is analyzed and compared with that of the Particle filter and the proposed system. The results show that the accuracy is improved marginally in a few data sets while in others it has enhanced significantly. In the data set 4 (Figure 20a), it can be seen that there are two cars on the extreme right are partially occluded. This partial occlusion is well detected and the handled using distance formulation technique wherein the links on the particle group of each car to the bus created, becomes zero or in other words the convergence factor of the distance across the two cars becomes zero. This indicates the presence of occlusion. In the Fig.23, illustrates the accuracy in tracking even during occlusion.

Figure 23: Tracking Results in Case of Partial Occlusion

The graph (Figure 23 b) shows the quantitative accuracy comparison of Particle Filter and that of the proposed system in case of Partial occlusion. In case of partial occlusion, one object is partially hidden by a background structure from its visibility through a single moving camera. It can be inferred that the accuracy levels are almost the same as that of the Particle filter with a marginal increase of about 2%. This is because the nature of the data set is a racing video where the cars are in either linear or non-linear motion with no sudden shifts in laps or uncertain illumination problems.
Inter-object Occlusion

In the data set 4 (Figure 24 a), it can be seen that there is occlusion between two foreground objects, that is two cars here. This occlusion is also well detected and handled using the distance formulation technique and superposition estimation. The Figure 24 a) illustrates the accuracy in tracking even during and after occlusion.

**Figure 24:** Tracking Results in Case of Inter-Object Occlusion

![Tracking Results in Case of Inter-Object Occlusion](image)

The graph (Fig.24.b) shows the quantitative accuracy comparison of Particle Filter and that of the proposed system in case of inter-object occlusion. In case of inter-object occlusion, one object is partially hidden by another from its visibility through a single moving camera. It can be inferred that the accuracy levels are having increased by 5% than that of the Particle filter. This is because the particle grouping and the distance formulation technique accurately keeps track of the object’s position and label.

Complete Occlusion between Moving Objects

In the data set 4 (Figure.25.a), it can be seen that there are two cars in the front that are in occlusion. There occurs a complete occlusion between the two cars that is one car completely hides the other car blocking its clear visibility from the camera point of view. This is well detected and handled using the superposition estimation technique based on energy computations of the individual object. Figure 25. a, illustrates the accuracy in tracking even during occlusion.

**Figure 25:** Tracking Results in Case of Complete Occlusion

![Tracking Results in Case of Complete Occlusion](image)

a) Data set 4 at varying times
Occlusion is detected when the two cars go missing from the scene and it is handled effectively once they reappear into the scene after crossing the bridge. This is handled using superposition estimation method and the distance formulation technique keep track of the object’s position and label continuously and accurately.

**Occlusion with Background Object**

In the data set (Figure 26), it can be seen that there are two cars that undergo occlusion. There occurs a complete occlusion when the two cars go under a bridge which is a stationary background object. Occlusion is detected when the two cars go missing from the scene and it is handled effectively once they reappear into the scene after crossing the bridge. This is handled using superposition estimation and entanglement free treatment based on energy computations of the individual object.

In this case, the distance formulation technique alone will not suffice in successfully tracking and labelling the objects because the objects’ position information is missing for a few fractions of seconds. Thus the motion and energy parameters of the objects are to be used in order to detect those objects once they arise out of occlusion from the background object.

**Figure 26:** Tracking Results in Case of Complete Occlusion by a Background Structure (Bridge)
The graph (Figure 26.b) shows the quantitative accuracy comparison of Particle Filter and that of the proposed system in case of complete occlusion of objects in the background. In case of complete occlusion, the objects are completely hidden by a background object thus completely blocking its visibility through a single moving camera.

**Speed Analysis of Different Resolutions for Multiple Objects Tracking**

**Table 6:** Speed Analysis for Multiple Objects Tracking

<table>
<thead>
<tr>
<th>Nature of the Dataset(MO)</th>
<th>Accuracy</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>80.2%</td>
<td>25.8%</td>
</tr>
<tr>
<td>L with more occlusion</td>
<td>75.8%</td>
<td>34.2%</td>
</tr>
<tr>
<td>L &amp;NL</td>
<td>81.2%</td>
<td>29.8%</td>
</tr>
</tbody>
</table>

**Table 7:** Accuracy of Proposed Model for Multiple Objects Tracking

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Video Resolution</th>
<th>Object size</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>480x360</td>
<td>142x60</td>
<td>8~13 fps</td>
</tr>
<tr>
<td>II</td>
<td>640x479</td>
<td>230x180</td>
<td>5~8 fps</td>
</tr>
<tr>
<td>III</td>
<td>640x332</td>
<td>290x110</td>
<td>5~8 fps</td>
</tr>
</tbody>
</table>

The speed of the proposed model was analyzed by subjecting the system to different datasets where the object size was variable in nature.

On an average it was found that the speed of the tracking (Table 6) in case of multiple objects was found to be 8 fps. This is comparatively 12.5% increase with that of the system described in [20].

The accuracy (table 7) of the proposed model for multiple object tracking for three different data sets on an average was found to be 80% but the missing detection rate were 28% on an average.

**VI. Conclusion**

The experimental results and the inferences show that the method proposed clearly estimates the information about the object region with the help of the machine aware system learning, particle grouping and bus topology approach. Though the nature of the particle filter allows for efficient processing of a non-linear system, implementation becomes quite slow for real time problems because of handling of symbolic constants and thus needs a lot of calculation time. As an example, when particle filter was applied to the data set 1, the average response time was found to be 2.31s while that of the proposed system was as less as 1.43s. The particle grouping technique in comparison with the traditional application of the particle filter has halved the number of particles that are required to be processed for tracking. For example, for a frame of resolution 350x240, a total of 2040 particles are required. But in the proposed system, due to grouping of particles it is reduced to 1050 particles which
is approximately 50% reduction in computation. On an average there is a 40% increase in performance of the system and it is proved to be effective under different aforementioned scenarios, while Kalman filter fails abruptly in tracking the objects under non-linear motion. Also the accuracy in tracking the moving objects under various types of occlusion has been enhanced by 30% due to the superposition estimation and distance formulation techniques in comparison with the other filters. Also Kalman filter has an appreciable false tracking rate which degrades the working of the system. Although Particle filter handles partial occlusion, its computational complexity increases exponentially with a need for a number of particles for processing. But on applying the proposed particle grouping method, the number of computations is almost halved as in the case of dataset 1.

VII. Future Work
As part of future work, we intend to track multiple kinds of same colored objects like car, bus, truck and lorry in the presence of occlusions. Also the automatic labeling of the objects that appear randomly in the scene could be worked upon and the system can be extended in a way to perform Rotation State Torsion (RST) independent tracking of objects that may undergo rotational changes when in motion. Segmentation of multi-targets in order to enhance the generality of the object detection engine could also be focused upon.

References


