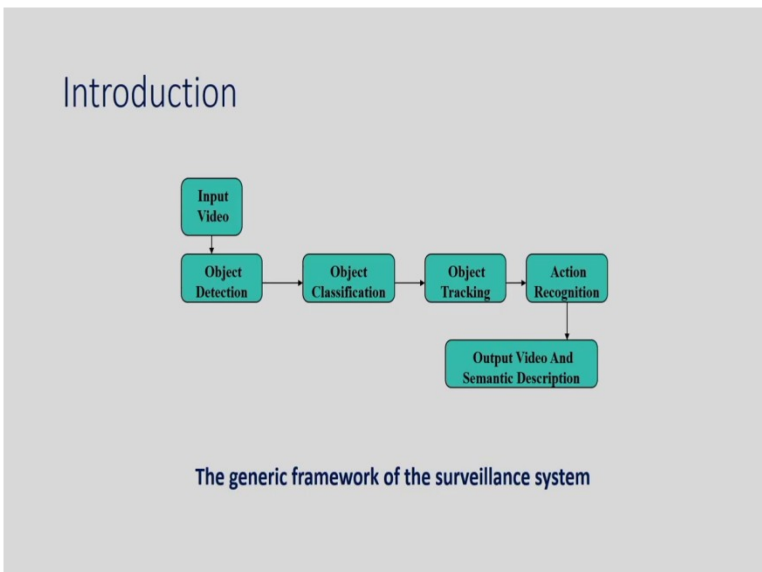
Computer Vision and Image Processing – Fundamentals and Application
Professor Doctor M. K. Bhuyan
Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati, India
Lecture 40
Introduction to Video Surveillance
(Object Tracking)

Welcome to NPTEL MOOCs course on Computer Vision and Image Processing - Fundamentals and Applications. In my last class I discussed the concept of background modeling and the concept of the motion estimation. For motion estimation, I discussed one important algorithm, that is optical flow based motion estimation. Today I am going to continue the same discussion. So, first I will explain the concept of the object tracking, after this I will briefly explain the concept of particle filter based object tracking. So, let us see what is object tracking?

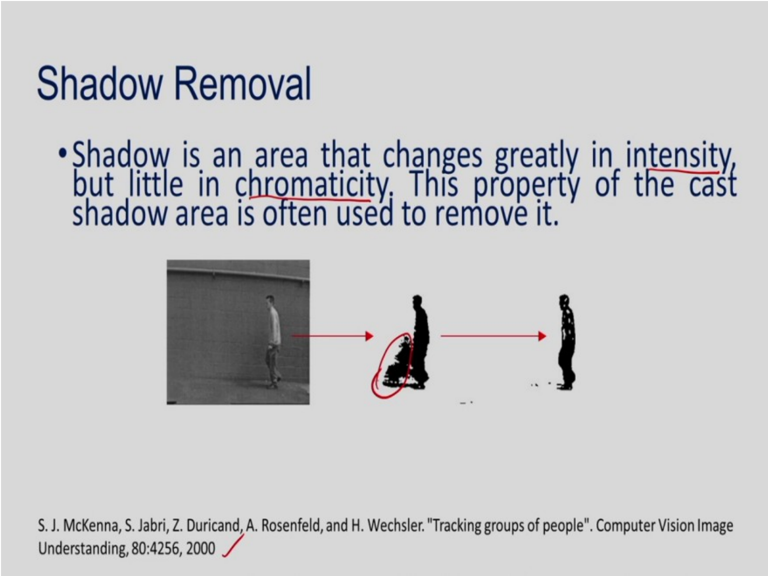
(Refer Slide Time: 01:07)



So, in my last class I discussed the concept of the video surveillance system. So, in this block diagram, I have shown the generic framework of a video surveillance system. So, you can see here the input video is available, that can be captured by the camera. After this we have to detect the objects present in an image or a present in the video. And after this we have to do the object classifications.

And after this we can do the tracking, the object tracking and after this the action recognition and the output is, the output video and semantic description. This is a generic framework of a video surveillance system. Last class I discussed this system.

(Refer Slide Time: 01:55)

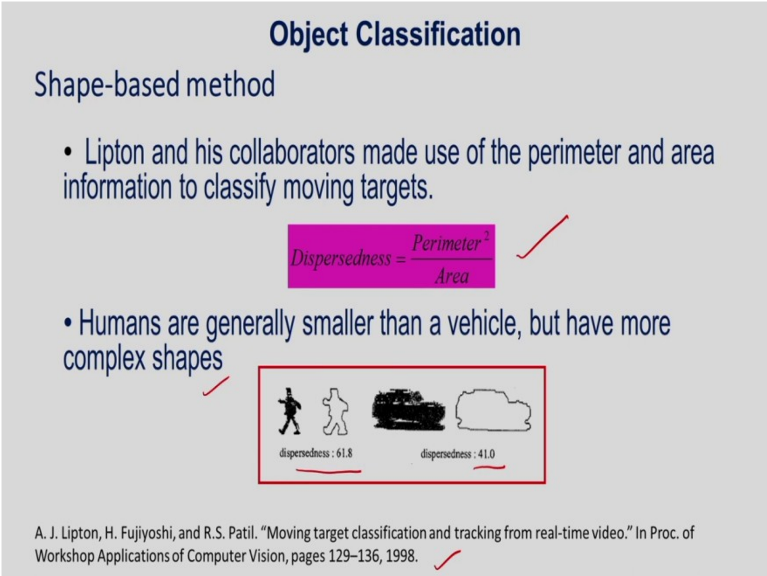


And one problem already I have mentioned, the problem of the shadow. So, we have to remove the shadow. So, if I consider the cast shadow, suppose if I want to do the tracking of an object, the shadow is also moving and the object is also moving. So, we have to remove the shadow. So, based on the motion information, we cannot remove the shadow. So, for this, we can consider some, a process here I have shown one classical approach in this paper.

So, in a shadow region, intensity changes greatly. However, change of chromaticity is negligible. So, based on this condition, we can separate the shadow from the object. So, in this example, you can see the cast shadow is present and based on this property that is the intensity variation and also the chromaticity variations, we can remove the shadow. But at present there are many new techniques for shadow removal.

So, this is the classical approach for shadow removal. There are many new techniques of shadow removal, even in the deep learning framework, we have different methods for shadow removal. So, you can see all these methods, but here I have briefly explained the concept of shadow removal.

(Refer Slide Time: 03:14)

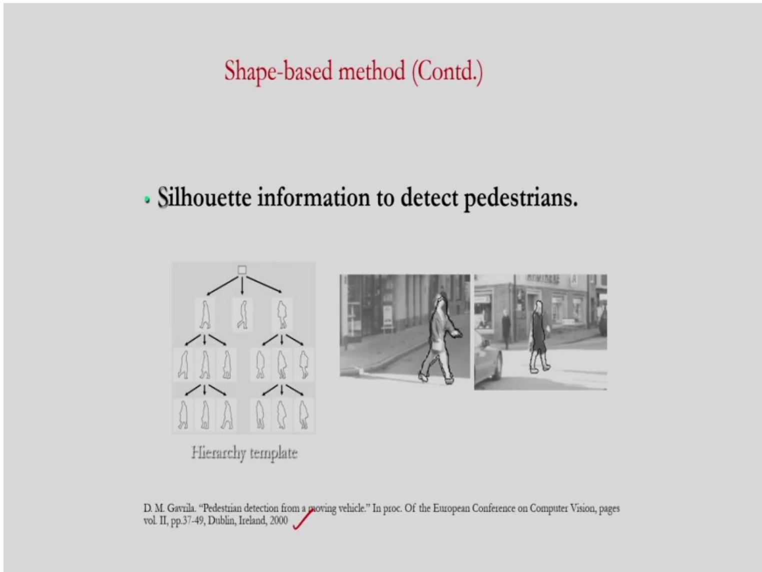


After this, the next point is the object classification. So, for these we can consider the shape feature for object classification. And here also I am considering a very old paper, but concept is very simple. Based on the shape information, we can classify human and the vehicles. So, for classification of humans and the vehicles, they considered one parameter, that parameter is dispersedness and it is based on the parameters and the area.

So, based on this parameter, they classified humans and the vehicle. So, for humans the value of this parameter is 61.8. And if I consider the vehicle the, the value is 41.0. The concept is, humans are generally smaller than the vehicles, but have a more complex shape. So, based on these parameters, we can do the classification, the classification of humans and the vehicles.

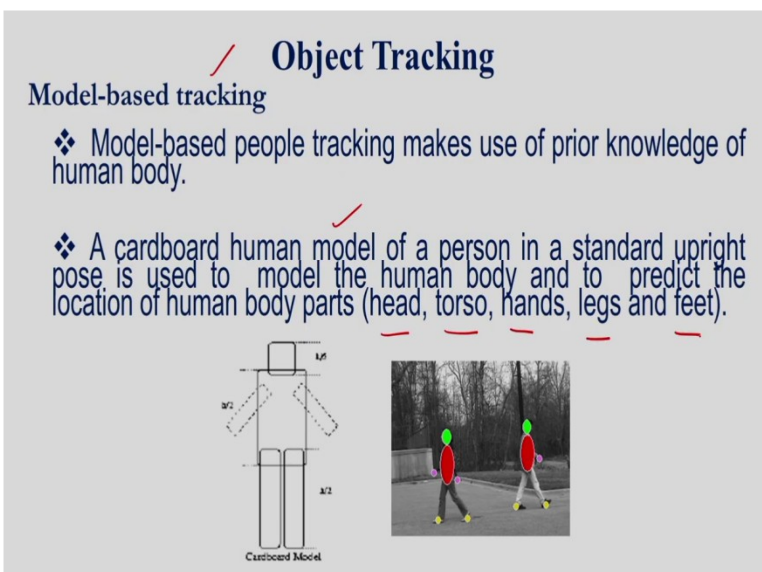
And this is also a very old technique, there are many new techniques like in deep learning also, we have many techniques for object classification, or maybe some other features based techniques we can consider for object classifications.

(Refer Slide Time: 04:31)



And this is another shape based method. The silhouette information can be considered to detect pedestrians. In this paper, they considered a hierarchy template. And based on this template, they detect the pedestrians. So, these are the templates and based on this template, they detect the pedestrians.

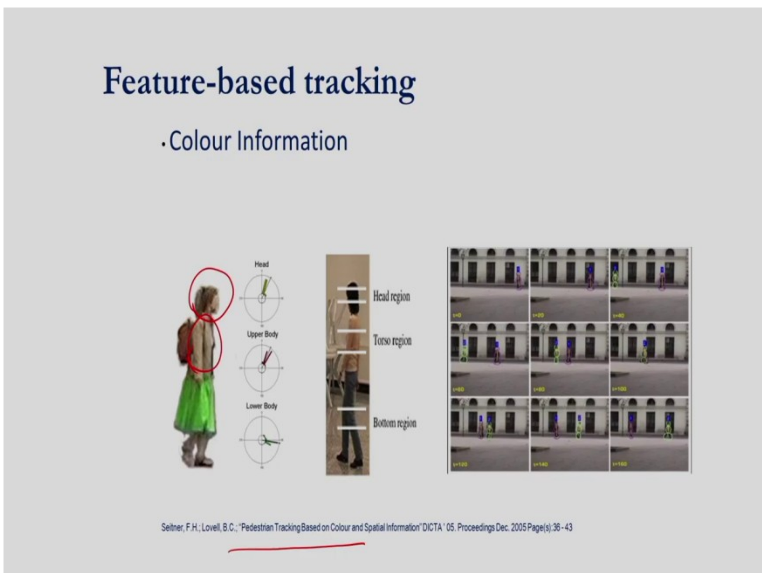
(Refer Slide Time: 04:50)



And after this we have to consider the object tracking. Tracking means finding the correspondence between the frames of a video. So, for this we can consider the model based tracking. Suppose if I want to track humans, then in this case, we can consider the model based tracking.

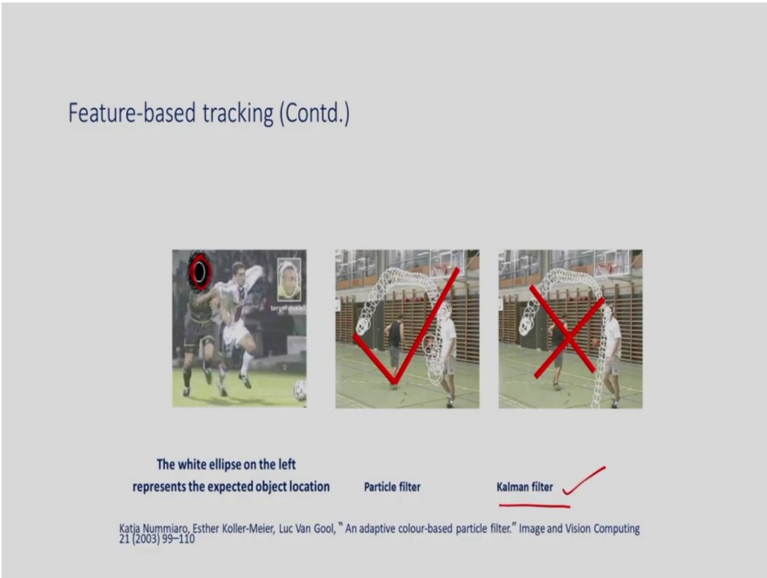
So, maybe we can consider a cardboard human model we can consider. And this model can be considered for object tracking, that is for modeling the human body and to predict the location of human body parts. The human body parts maybe the head, torso, hence, legs and the feet. So, this is the cardboard model, that is the cardboard human model and based on this model, we can do the object tracking.

(Refer Slide Time: 05:36)



And also we can consider the feature based tracking. In this paper, they considered the color information, based on the color information they did the object tracking. So, in this example, what have they considered? They considered a color histogram. So, color histogram corresponding to this portion, the head portion, color histogram corresponding to the upper body and color, color histogram corresponding to the lower body, based on the color information they did the tracking, that is the feature based base tracking.

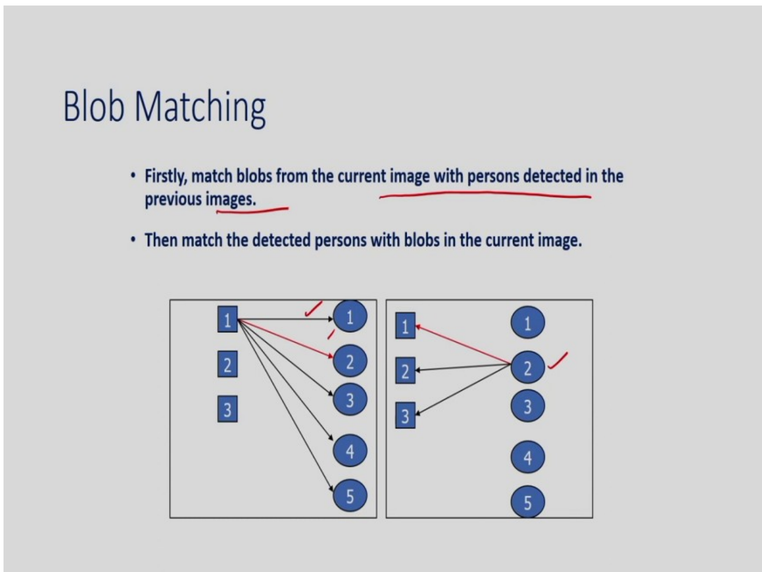
(Refer Slide Time: 06:06)



Here you can see I am showing one example of feature based tracking, the tracking of the ball. So, for these there are two popular algorithms, one is the Kalman filter, another one is the particle filter. In case of the Kalman filter, one assumption is the linear assumption, that concept I will explain later on. And also the noise is considered as Gaussian noise. So, because of this the Kalman filter cannot track properly, if the motion of the object is very high, then in this case it fails.

So, the advanced version is the particle filter, it considers a nonlinear state equation that I will explain later on, also the noise is the non Gaussian noise. So, that is why the particle filter can track objects very nicely, if you see in this example, the particle filter can track the objects very nicely as compared to the Kalman filter. And I assume that you know the concept of the Kalman filter, because today I am going to explain only the particle filter. So, you have to read the concept of the Kalman filter.

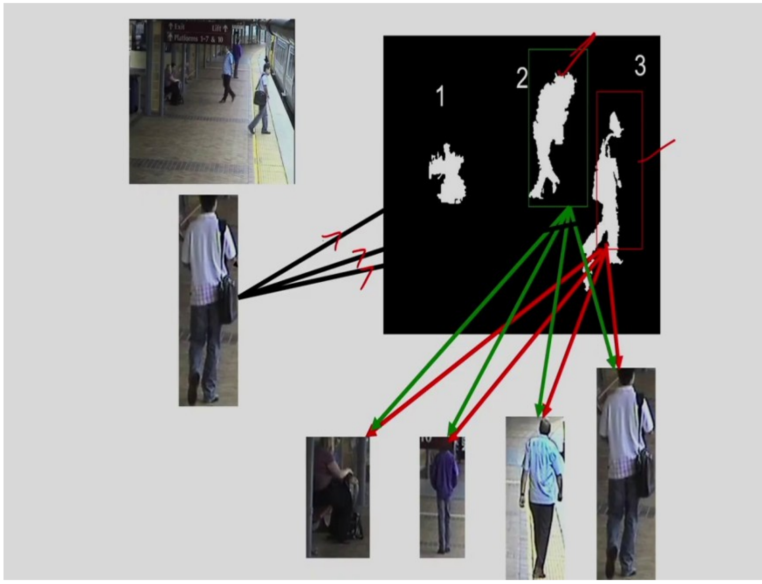
(Refer Slide Time: 07:10)



And one tracking algorithm is the blob matching. So, here you can see after background modeling, we will be getting the foreground objects, that these are the blobs. So, first, what do we have to do? Match blobs from the current image with a person detected in the previous image. So, that means, just I am matching the blobs of the current image, with persons detected in the previous image. So, this is the matching, after this I am getting the matching.

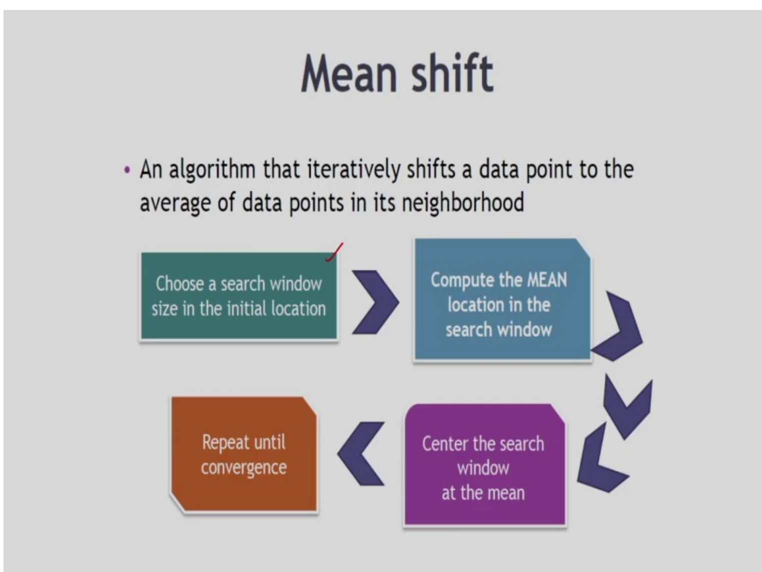
In this case the matching is two, then the match the detected person with the blobs in the current image. So, that is nothing but the two way matching. So, by using this two way matching, we can do the tracking. So, in the next slide you can see this concept, what is the two way matching and that is nothing but the blob matching, the blob means that is the foreground objects detected after background modeling.

(Refer Slide Time: 08:06)



So, here you can see, that I am considering the tracking of these pedestrians, and you can see I am detecting the blobs, these are the blobs after background modeling. So, I am getting the foreground objects. So, first I have to do the matching. So, just I am doing the matching. After this, after matching I am doing the reverse matching. So, suppose this person is detected and after this again I am doing the reverse matching, that is nothing but the two way matching. So, that is a very simple algorithm, that algorithm is the blob matching algorithm.

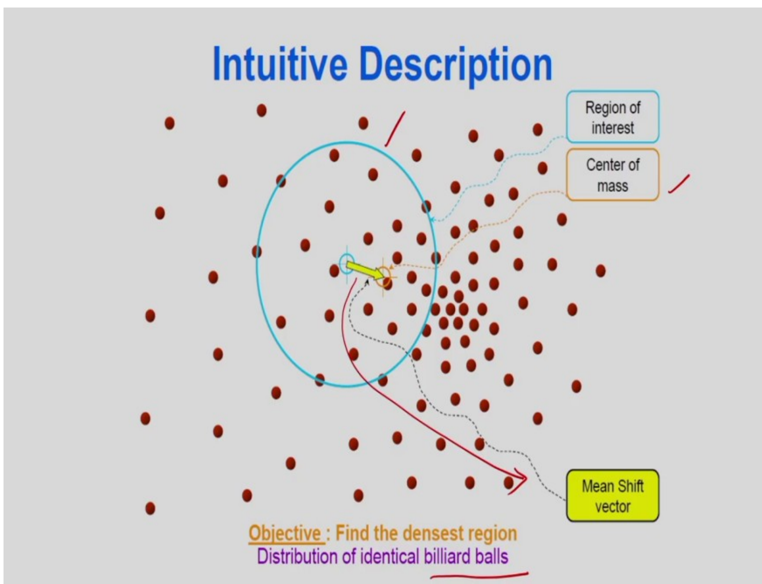
(Refer Slide Time: 08:40)

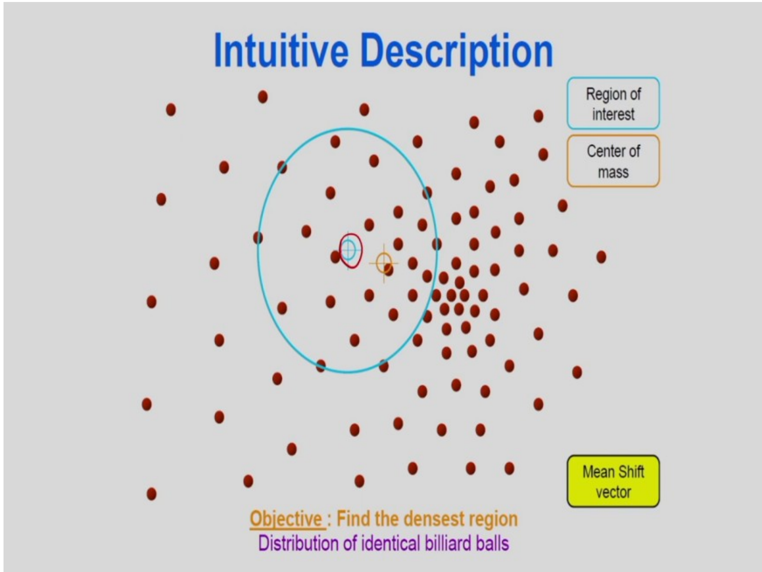


After this I will consider another algorithm for object tracking that is the mean shift algorithm. So, what is the objective of the mean shift algorithm? An algorithm that iteratively shifts a data point to the average of data points in its neighborhood. So, for this is what we have to consider? We have to consider a search window, we have to consider the initial location. After this we have to compute the mean location in the search window, that we have to compute.

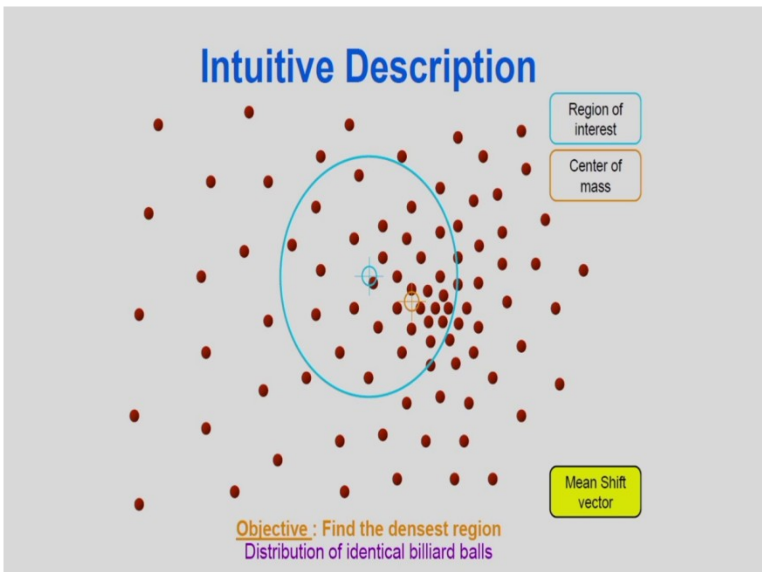
After this the center, the search window at the mean. So, that means, we have to reposition the search window at the new mean. And we have to repeat this until we get the condition of convergence. So, this is the concept of the mean shift algorithm. So, pictorially this concept I can explain in my next slide.

(Refer Slide Time: 09:31)

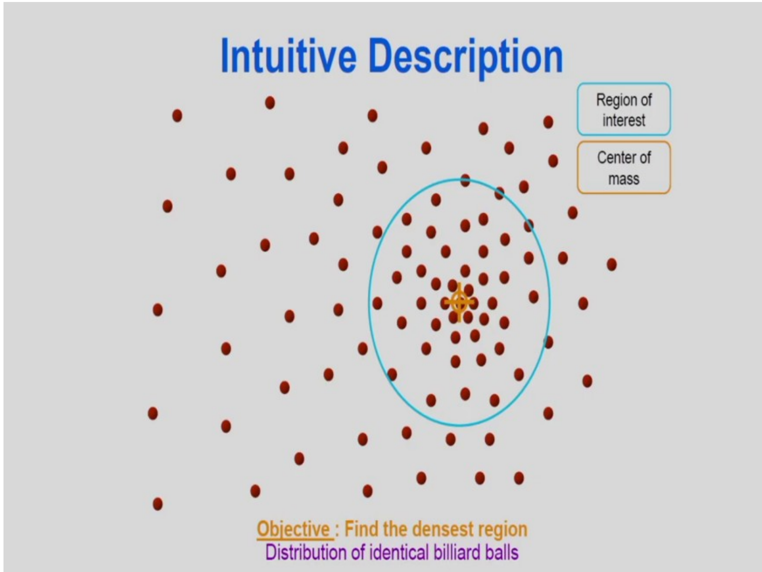




Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls



Intuitive Description
Region of interest
Center of mass
Mean Shift vector
Objective : Find the densest region
Distribution of identical billiard balls



Here I am showing some of the sample points, but here I am considering the distribution of the billiard balls. But let us consider it as the sample points. And this blue circle, it corresponds to the window that is the region of interest. And corresponding to this region of interest, we can determine the center of mass. So, the center of mass corresponding to these sample points, within the window, that we can determine. And you can see the mean shift vector, that is the vector from the center of the window and a new centroid that is the center of mass.

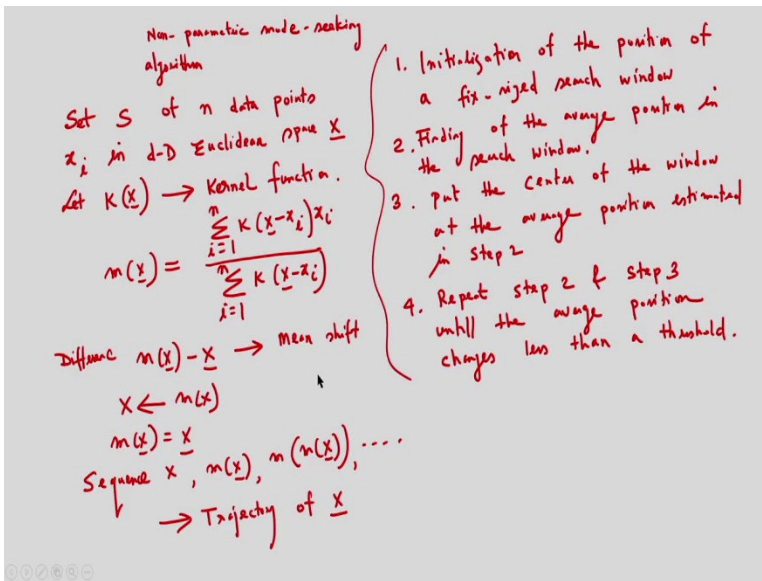
So, you will be getting the mean shift vector you will be getting. That is the man shift vector. I have to move the center of the window to the new center of the mass. So, that means, you can see, this is nothing but the center of the window, that is the region of interest. And after this, we have determined the center of the mass. After this, we have to move the center of the window to the new center of mass.

So, that means I am doing this, that means I am moving the center of the window to the center of the mass. After this, I am re centering the window, I am re centering the window and again I am calculating the center of the mass. And after this again I have to shift the center of the mass to the new mean. So, corresponding to this case also we can determine the mean shift vector, we can determine.

And again we have to seek the center of the mass to the center of the mass, that I am doing the shifting, that is the mean shift vector we can determine. And you can see, I am getting the new position of the center of the mass. And finally, we have to do this process iteratively and you can see this is the point of convergence. So, I am getting the center of the mass, that is nothing but the center of the window, the search window.

So, this is the, this is the pictorial representation of the mean shift algorithm. So, I am explaining briefly what the mean shift algorithm is. So the mean shift already I have explained, it is a nonparametric mode seeking algorithm.

(Refer Slide Time: 11:56)



So I can say it is a nonparametric, nonparametric mode seeking algorithm. So, I have to determine the mode, mode seeking algorithm for a density function. And this means shift algorithms iteratively shift a data point to the average of data points in each neighborhood. And this concept is very similar to clustering. Suppose let us consider a set S, set S is considered, set S is considered of n data points, of n data points. The data points are x_i suppose, in suppose d dimensional Euclidean space.

And I am considering the vector, the vectories X. And suppose let K_x is the kernel function. So I am considering the kernel function. This kernel function indicates how much x contributes to the estimation of the mean. Then, then the sample mean m at x with kernel k can be determined like this. So, that means I can determine the sample mean, the sample mean is m × I can determine. With a kernel, the kernel is k. So, already I have defined the kernel, the kernel I am considering.

The sample mean I can determine like this, i is equal to 1 to n because n number of data points we are considering. So, I can determine the mean, the that is the sample mean I can determine. And the difference, m x- X. So, this difference is called a mean shift. This difference is called a mean shift. So, mean shift algorithm iteratively moves data point to its mean, that concept already I have shown in the previous slide.

And each iteration, the mean mx is assigned to X that means, at each iteration that is mx is assigned to X. The algorithm stops when m x is equal to X. So, this is the convergence condition. That means the algorithm stops when mx is equal to X. So, that means, I will be getting the sequence, the sequence X is nothing but mx. After this I will be getting m, mx. So, like this I will be getting.

So, these are nothing but the trajectory of X, so these are nothing but the trajectory of X. And if the sample means are computed at multiple points then f is iterations, update is done simultaneously to all these points. So, this is the basic concept of the mean shift algorithm. For object tracking we can apply the mean shift algorithm. That is nothing but the kernel based tracking. For this what we can consider?

The appearance of an object can be characterized using histograms. And tracking can be done based on these histograms. But, one problem is that, that is, it is hard to specify an explicit to the parametric motion model to track non rigid objects. So, suppose like a walking person, that is a non rigid object. So, for these appearances of the non rigid object can sometimes be model which color distributions.

So, maybe we can consider the color distribution for object tracking. That means, the color histogram we can consider for the non rigid objects, like, like a walking person. So, briefly, what are the steps of the mean shift algorithm? The first step is, the initialization, initialization of the position of a fix size search window. So, this is the first step. The second step is finding the average position in the search window, number two.

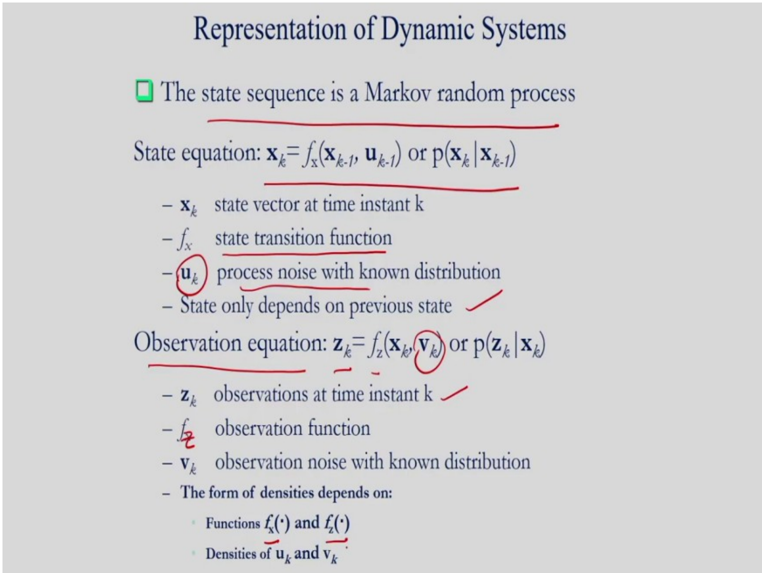
So, finding the average position in the search window. Number three0 we can consider, put the center of the window, put the center of the window at the average, average position estimated, estimated in step two. And finally, repeat step two and step three, step two and step three until the average position, until the average position changes less than a threshold. This is the mean shift algorithm, we can do the tracking.

So first I have to do the initialization of the portion of a fix size window. That is the search window, after this the finding of the average position in the search window and after this put the center of the window at the average position estimated in step number two, repeat step number 2 and step number three, until the average position change is less than a threshold. So brief, this is the brief concept of that mean shift algorithm. So, with the mean shift algorithm you can do the tracking.

(Refer Slide Time: 20:21)

After this, briefly I will explain the concept of the recursive filter. And already I have explained that, the particle filter can track objects perfectly as compared to the Kalman filter. And I assume that you know the concept of the Kalman filter in this discussion.

(Refer Slide Time: 20:40)



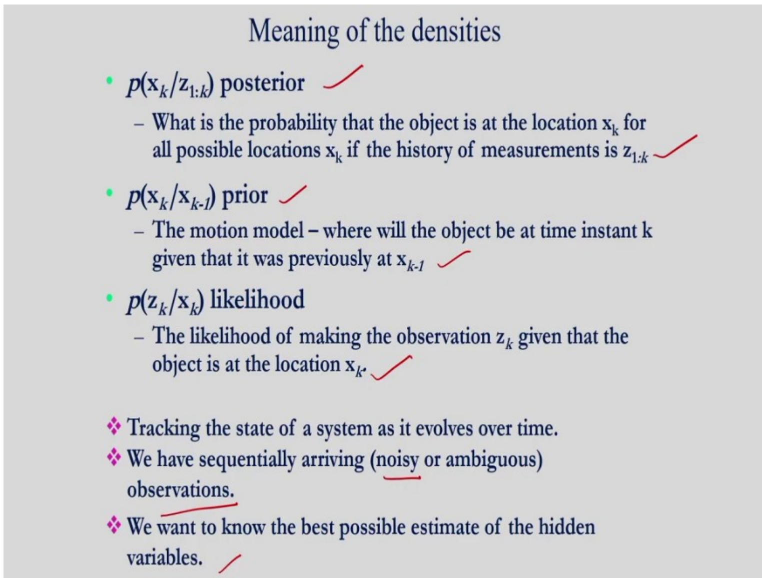
So, in case of the recursive filter, the first one is the representation of dynamic systems. So, first the state sequence is a Markov random process, that is the condition. The state sequence is represented by this. So, $x_k$is the state vector at the time k. And I am considering the function, the state transition function f_x is the state transition function, that is considered and I am considering the process noise of the known distribution that is u_k. And the state only depends on the previous state, that is the concept of first order Markov process.

So, here you can see x_k depends on x_k- 1, that is the state only depends on the previous state. And in this case what I am considering? u_k. -1, that is nothing but the process noise at time instant k minus 1. Here you can see p x_k given x_k- 1 that is nothing but the first order Markov process. And similarly, we have the observation equations. So, that is represented by$z_k f_z$ x_k, v_k.

So, for this what we have considered? z_k is the observation at times instant k. And again we are considering the observation function, the observation function is f_z. So, it should be f_z, that is the observation function. And v_k is the observation noise with known distribution. So, I am considering the observation noise v_k And in this case I am considering the process noise u_k that is the process noise.

And I am considering the observation noise with known distribution. And the form of density depends on these two functions, one is f_x another one isf_z. And we have the densities, one is the process noise u_k and another one is the observation noise. That is the representation of a dynamic system.

(Refer Slide Time: 22:46)

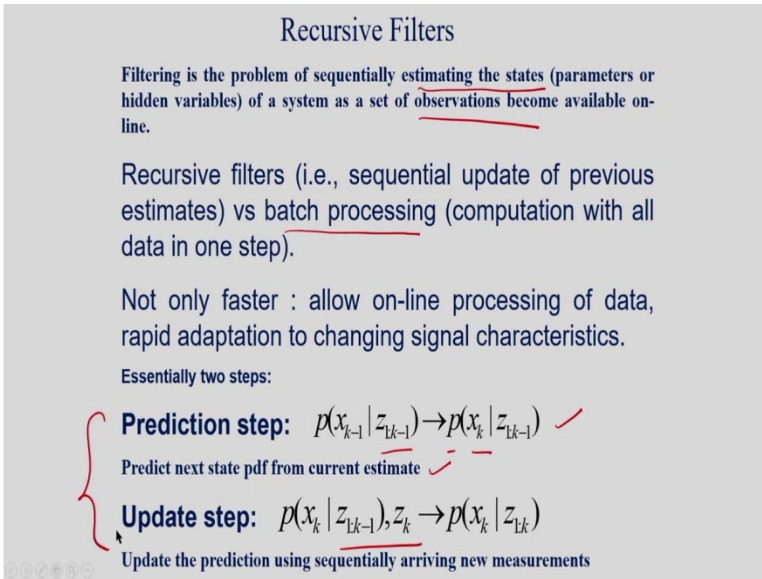


So, what are the meanings of the densities? The first one is the probability of x given z, one. That is the posterior density, what is the probability that object is at the location x_k? For all possible locations x_k?, if the history of the measurement is $z_1 : k$. So, this is the definition of the posterior density. And another one is the prior density, what is the prior density? Probability of x_k given $x-1$, that is the motion model.

Where will be the object be at the time instant k given that it was previously at$x_k - ¿$ 1, that is nothing but the first order Markov process. So, that is the prior information. Also we have the likelihood information, that is the plus conditional density. That is the likelihood of making the observations z_k, given that the object is at the location x_k. So, that is the likelihood. And what is the tracking, what is the definition of the tracking?

In this case, tracking the state of a system as it evolves over time. And we have sequentially arriving observations. That means, the noisy observations are available. So, from these observations, we have to do the estimation of the states, that is nothing but the tracking. So, we want to know the best possible estimate of the hidden variables. So, we have the observations and the noisy observations. And from these observations we have to predict the state, that is the tracking the state of a system as it evolves over time. So, that is the meaning of the tracking.

(Refer Slide Time: 24:36)



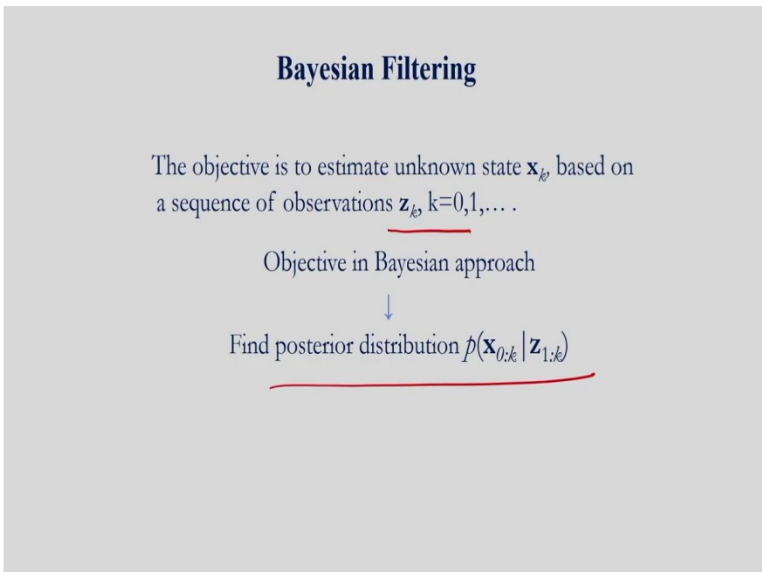
And in case of a recursive filtering, the filtering is the problem of sequentially estimating the states. That means, the parameters or the hidden variables of a system as a set of observation became available online. So, that means we have the observations and from these observations we have to estimate that states of a system. That is the filtering is a problem of sequentially estimating the states, that is nothing but the parameters or the hidden variables of a system. And based on the observations.

So, a recursive filter, that is the sequential update of previous estimate. But if I consider the batch processing computation with all the data in one step. So, that is the difference between the recursive filter and the batch processing. That is, in case of a recursive filter that is not only faster, but allows online processing of data repeat adaptation to changing signal characteristics. So, in case of the recursive filter, there are two steps.

One is the prediction step. That means, we have to do the prediction. So, from p x_k -1 and observation is available, we have to determine the probability x_k. So, from the given observation, we have to determine the probability of x_k that is the, current state we have to determine from the previous state, the previous state is x_k -1. And after this, that means, what is the prediction?

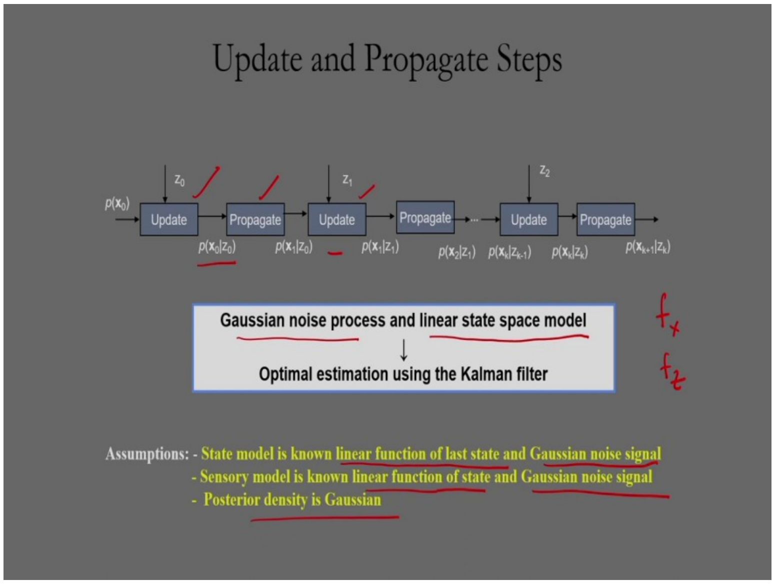
The predict the next state pdf from the current estimate, that is the prediction. After this we have to do the update, that is the update step. So, update the prediction using sequentially arriving new measurements, because we have the measurements, we have the observations. So, from these observations, we have to update the prediction based on sequentially arriving new measurements. So, that means, we have two steps. One is the prediction step, another one is the update step.

(Refer Slide Time: 26:39)



So, that is the concept of the Bayesian filtering. The objective is to estimate unknown state Xk based on a sequence of observations, the observations are Zk. So, we have the observations and we have to find a posterior distributions. So, we have to find a posterior distribution, that is the objective of the Bayesian approach.

(Refer Slide Time: 27:00)



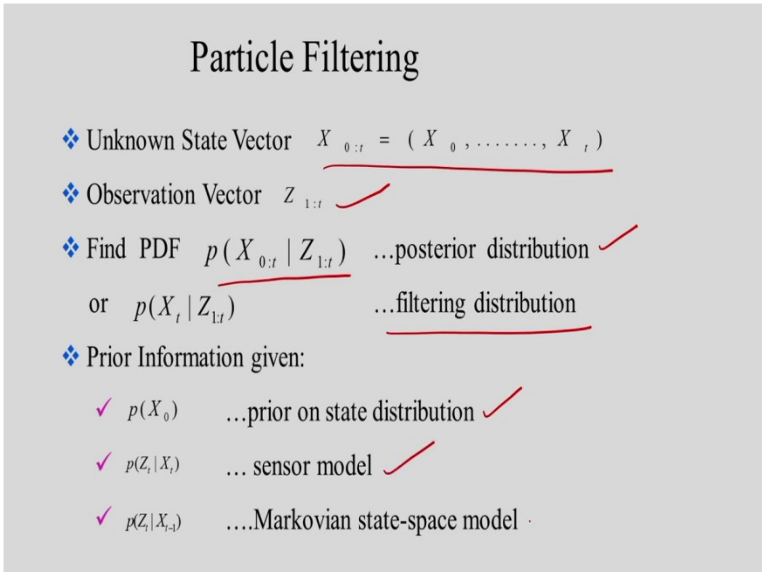
So, this concept I have shown pictorially here. So, you can see, first I am doing the updation, after this we are propagating. Again we are updating based on the observations, the observations are z0, z1, z2 like this. So, we are just predicting and after this again we are updating. We are doing the prediction and after this again we are updating, that is nothing but updating and propagating, updating and propagating.

So, this is the concept of the Bayesian estimation, that is the recursive filter. And in case of the Kalman filter, we have considered the Gaussian noise process and the linear state space model. So, if you see this function, the function is fx, another function was the fz. So, that is the linear function, that is considered in the Kalman filter. But, you can see, in case of the Kalman filter, we consider this the linear function of the last state and Gaussian noise signal, we have considered.

And the sensory model is nonlinear function of state, and the Gaussian noise signal we have considered. And the this posterior density is also Gaussian. So, these are the assumptions of the Kalman filter. Practically these assumptions are not true. So, if I want to track an object suppose, which moves very quickly, then these assumptions generally fails.

So, that is why the Kalman filter cannot track objects properly, that is the problem of the Kalman filter. So, that is why we may consider the particle filter. In particle filter these functions, these function we consider as nonlinear functions. And the noise we can consider as the non Gaussian noise.

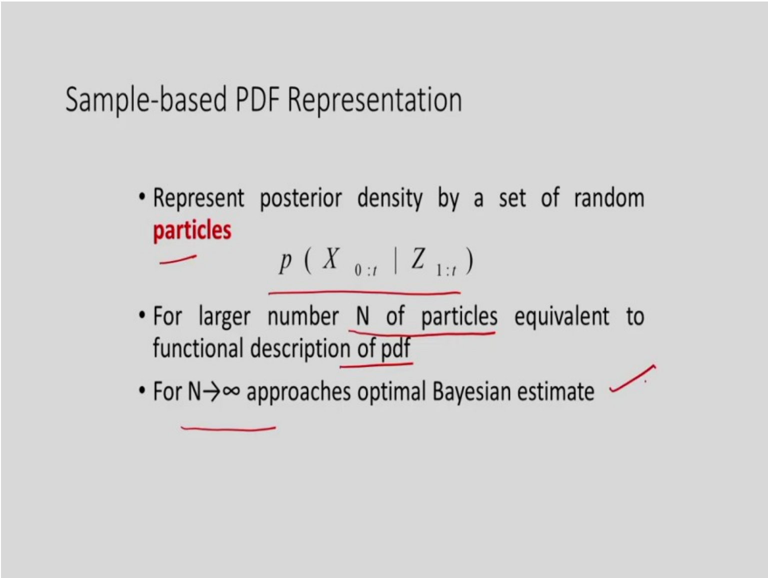
(Refer Slide Time: 28:46)



So, what is a particle filter? In particle filter unknown state vector is available. So, x0 to t, so all the states are available x0, x1, xt all these states are available. We have the observation vector Z from 1 to t. And we have to find a posterior distribution. So, this distribution we have to determine, that is the problem of the tracking. So, that means, we have to determine this one, that is the filtering distribution we have to determine from the observation.

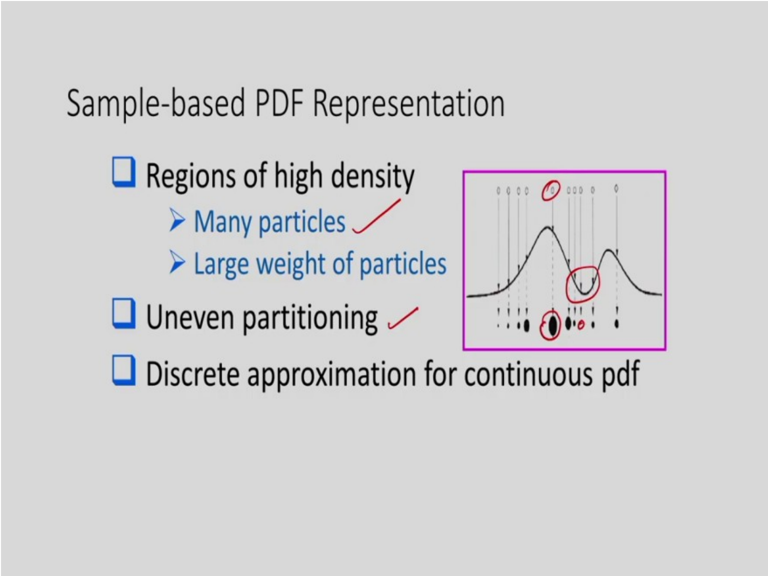
And prior information is given, that is the prior on the state distribution it is given. The sensor model is given and we have considered a Markovian state space model, that we have considered. So, this information is available.

(Refer Slide Time: 29:33)



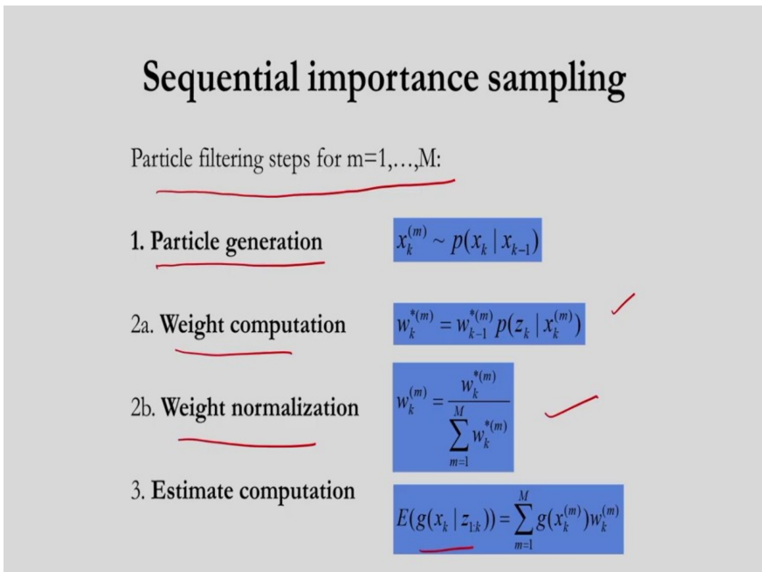
Now, in case of the particle filter, that posterior density is represented by a set of random particles. So, we have to consider random particles to represent the posterior density. If I consider large number of particles, suppose n number of particles equivalent to functional description of the pdf, the pdf is the posterior density. And if I consider N tends to infinity, then this particle filter process is the optimal Bayesian estimate. That means, it process optimal Bayesian estimate.

(Refer Slide Time: 30:10)



So, here you can see, I am showing the, the concept of the particle filter. Sample based PDF representation. The PDF is represented by random particles, regions of high density we are considering many particles. And also the large weight of particles and here you can see uneven partitioning. So, region of high density we are considering. Suppose, so, more number of particles we are considering. and here you can see corresponding to this we are considering the large weight. So, the weight of this particle is large as compared to this. And this is nothing but the discrete approximation of continuous PDF.

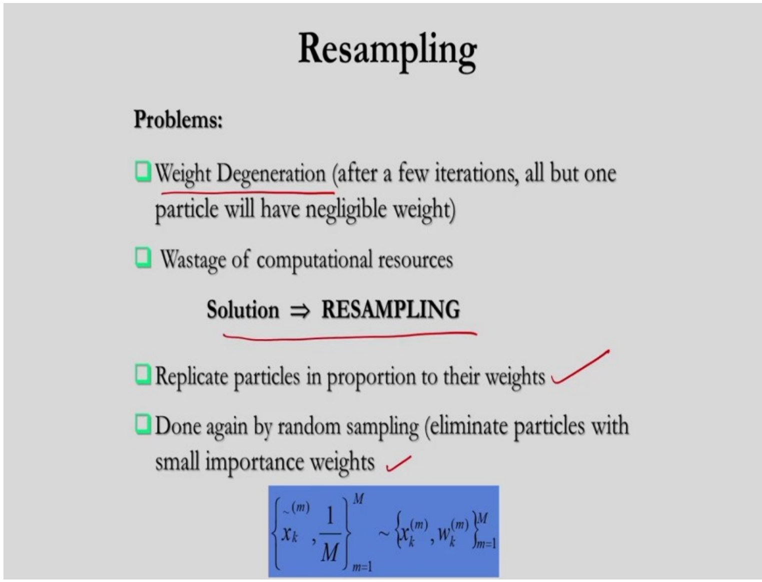
(Refer Slide Time: 30:52)



So, what are the steps of the particle filter? That is the sequential importance sampling, we have to consider. So, first the particle filtering steps for m is equal to 1 to M. So, first we have to define the particles, that is the particle generation. After this we have to compute the weight for the particle. So, for each and every particle we have to assign the weights, and also we have to normalize the weights, the weight normalization we have to do by using this equation.

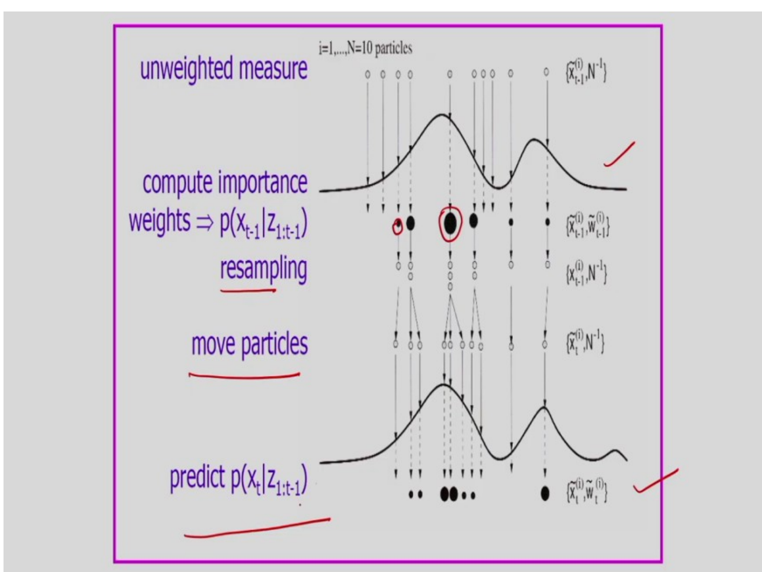
And finally, we have to estimate the states, that is the estimate computation. So, we are estimating the states So, these are the steps, that is the first one is the particle generation. After this we have to assign the weights for the particles, and after this weight normalization, estimate computations.

(Refer Slide Time: 31:42)



But what is the problem? After a few iterations, all but one particle will have negligible weight that is called the weight degeneration problem. So, for this we have to do the resampling, the solution is the resampling. Replicate particles in proportion to their weights, then again by random sampling, and we have to eliminate particles with small importance weight. That means, the particle having the negligible weight, that we have to neglect. That is called the importance sampling. So, this concept I am going to explain in my next slide.

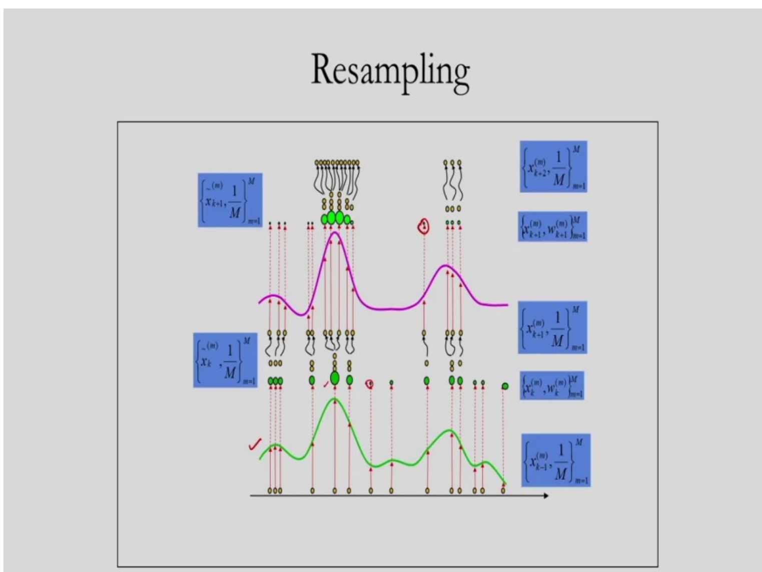
(Refer Slide Time: 32:22)



So, you are, here you can see, this pdf is approximated by the n number of random particles. And you can see, I am assigning the weights corresponding to all these particles, these are the weights you can see, and this particular has the maximum weight as compared to this particle. And after this we have to propagate these particles. So, move the particles and before the moving the particles, we have to do the resampling, resampling of the particles, move the particles.

And after this we have to do the prediction. And this is the state of the particle filter algorithm. So, that means the pdf is approximated by random particles. And also we have to determine the importance weight, we have to determine. And after this we have to do the resampling and after this we have to propagate the particles that is the move the particles and after this the prediction of the Xt given the measurement, the measurement is z1 to t minus 1. So, based on this we have to do the prediction.

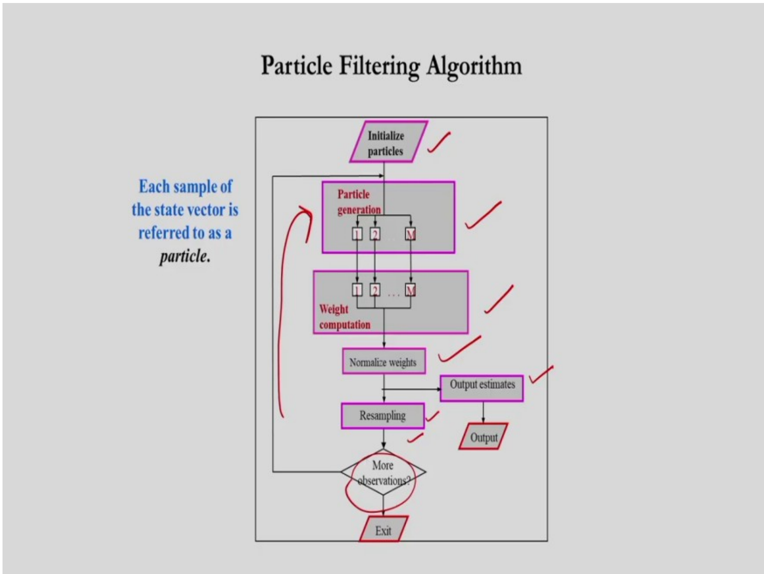
(Refer Slide Time: 33:29)



And the concept of the resampling because, after some iterations, some of the particles will have negligible weight, that we have to eliminate. So, here this concept, the concept of the resampling I am showing here. The pdf is approximated by the particles, the random particles and the associated weights. So, these are the weights and you can see some of the particles have negligible weights, that we can neglect.

And after this we can propagate the particles that means, we can move the particles to the next step. And like this, you can see in the next step, this particle has negligible weight that we can eliminate and like this we can move the particles into the next step, in the next iteration like this we have to do. And we can also the predict the states, based on the measurements. So, this is the concept of resampling.

(Refer Slide Time: 34:18)

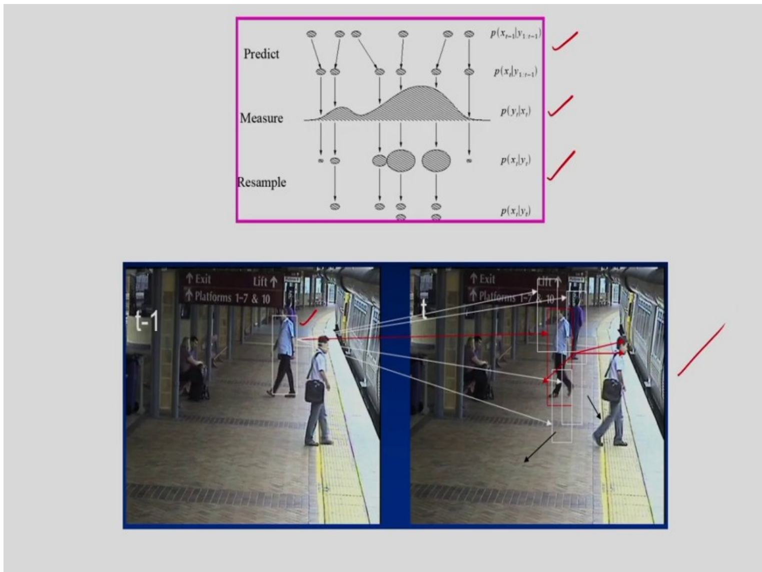


So, briefly I can say this is the particle filter algorithm. First randomly I have to initialize the particle, that is nothing but the discrete approximation of continuous pdf, that is the pdf, that is the posterior pdf is approximated by random particles. So, that is the particle generations. And after particle generation, we have to compute the weight of the particles by using that mathematics.

After this we have to normalize the weights. After this we have to resample the particles. So, we are getting observations and based on these observations, we have to do the tracking. So, we have to do the tracking like this. In this case, from the normalized weight we can estimate the output, that output estimate we can do, but the problem is that after some iterations, some of the particles will have negligible weight.

So, that is why resampling is important. So, we are doing the resampling and after this we are doing the particle generation, weight computation, normalized weights and after this we can output the estimates. And based on the observation we have to do the tracking. So, this is the particle filter algorithm. So, briefly I have explained the concept of the particle filter, but I think you have to study the research papers to understand the concept of the particle filter in more detailing.

(Refer Slide Time: 35:39)

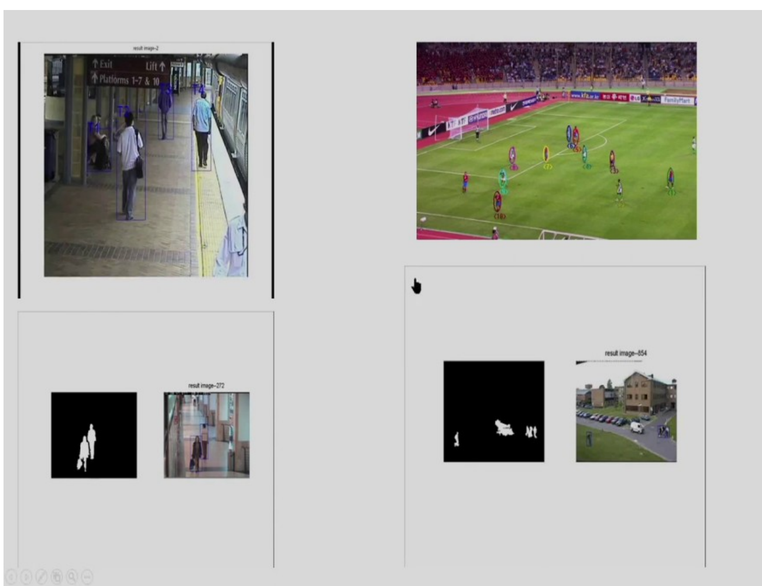


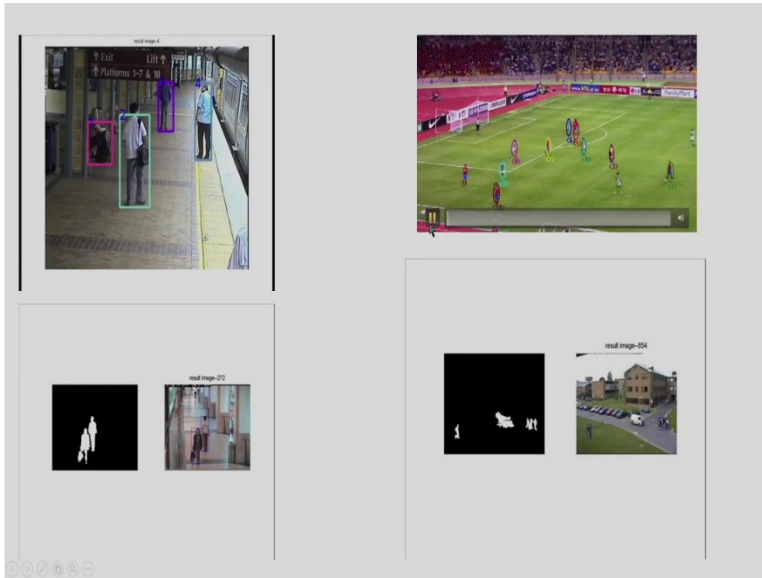
So, here you can see I am showing the tracking, the tracking of this person, tracking of this. So, based on the particle filters. So, first one is the prediction, again after this the measurement, after this the resampling, the resampling concept already I have explained. Because we have to neglect the, the particles having negligible weights, that we have to neglect. And after this, this we have to do the iterations. So, based on this we can do the tracking. So, this is one example of the tracking.

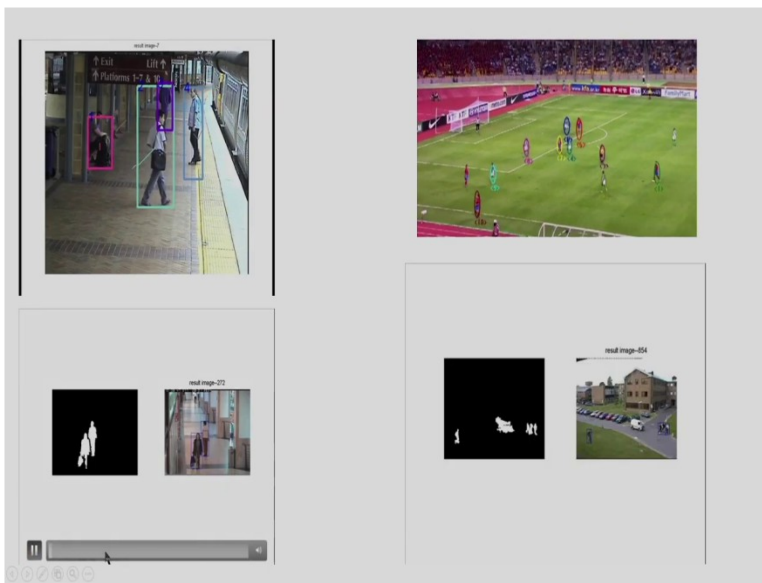
(Refer Slide Time: 36:13)

And this is also another example of the tracking you can see, the tracking.

(Refer Slide Time: 36:17)

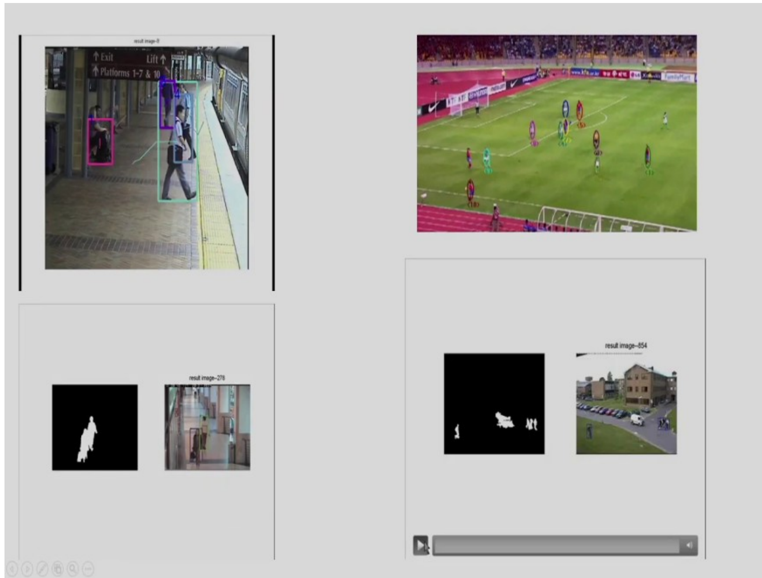


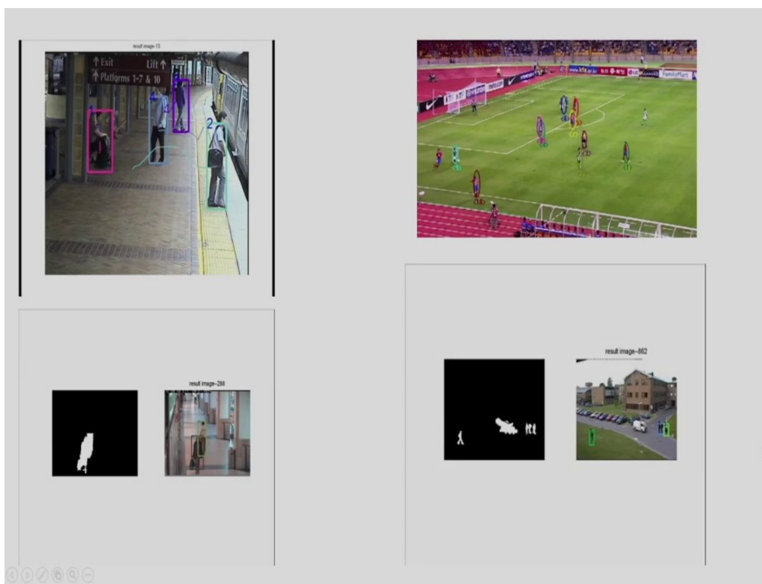


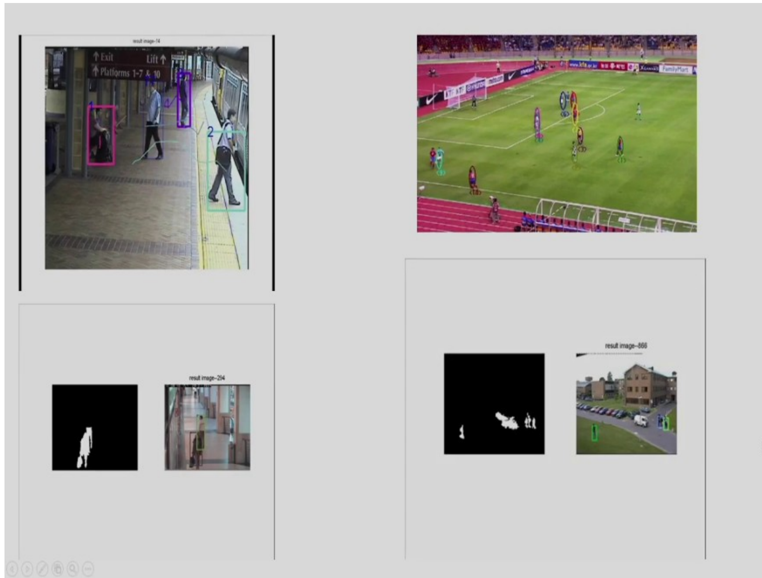


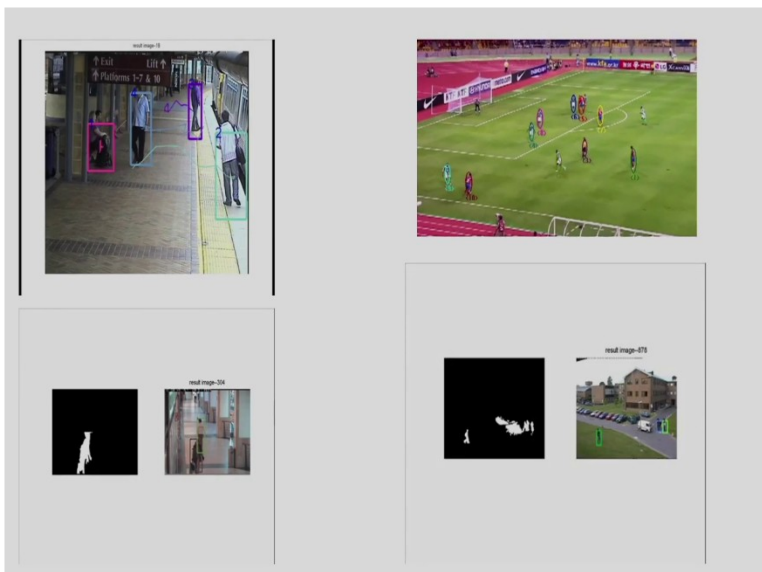
And I can show some of the videos of tracking. So, we have developed this algorithm the particle filter based algorithm for object tracking. So, you can see these videos.

(Refer Slide Time: 36:27)



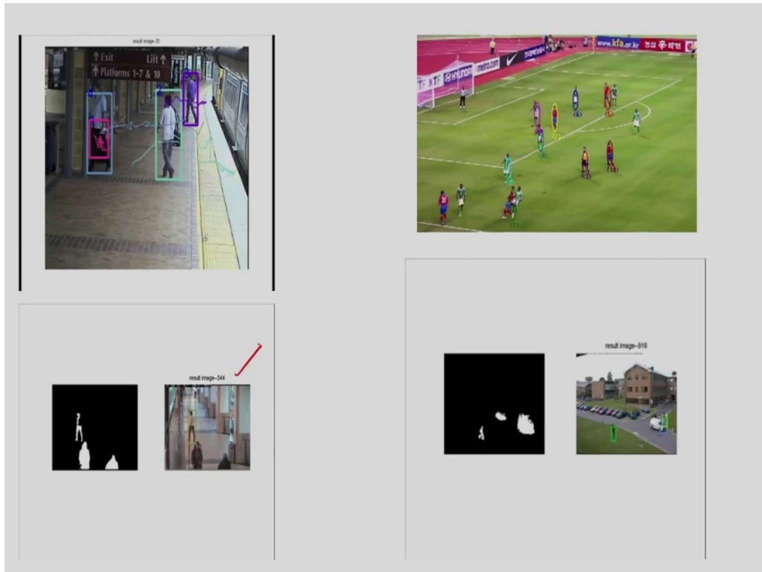


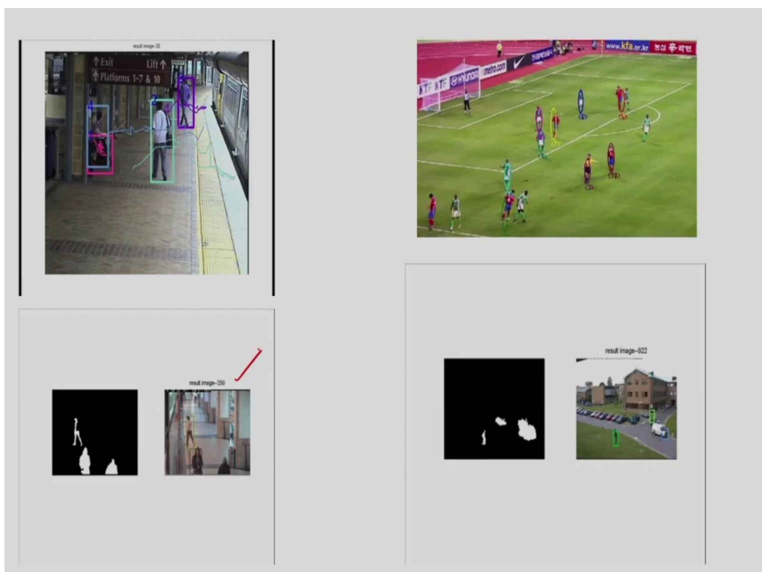


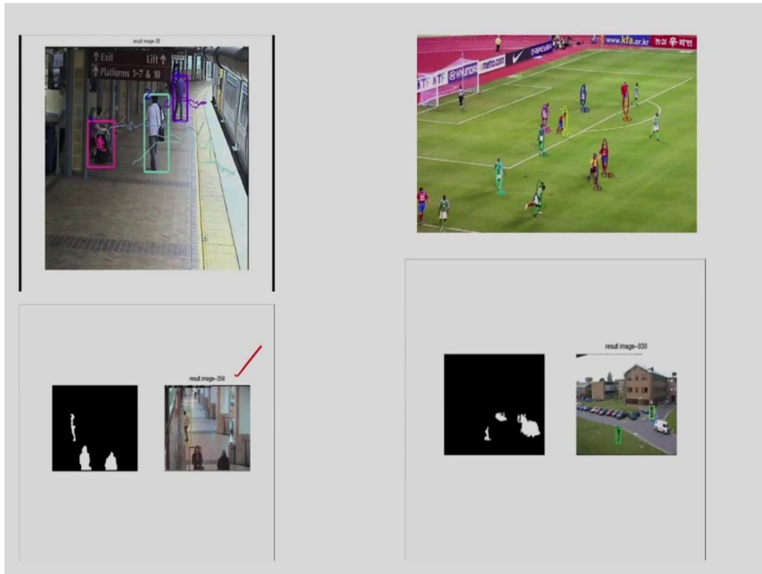


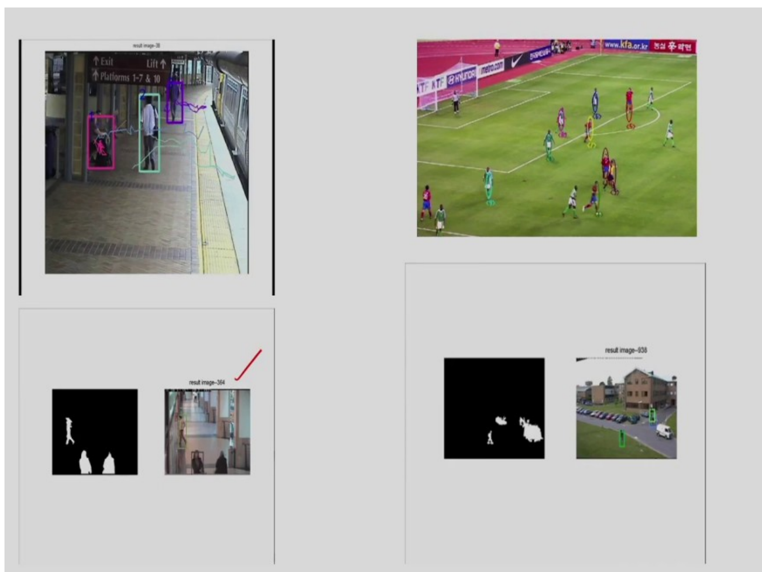
So, we have done this particle filter based tracking and also we have proposed some, the background modeling algorithms. So, based on this we have done the tracking.

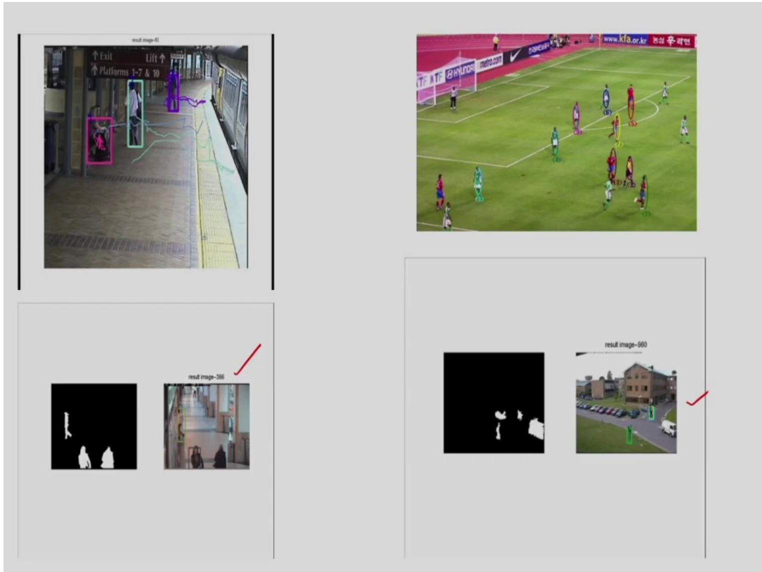
(Refer Slide Time: 36:41)





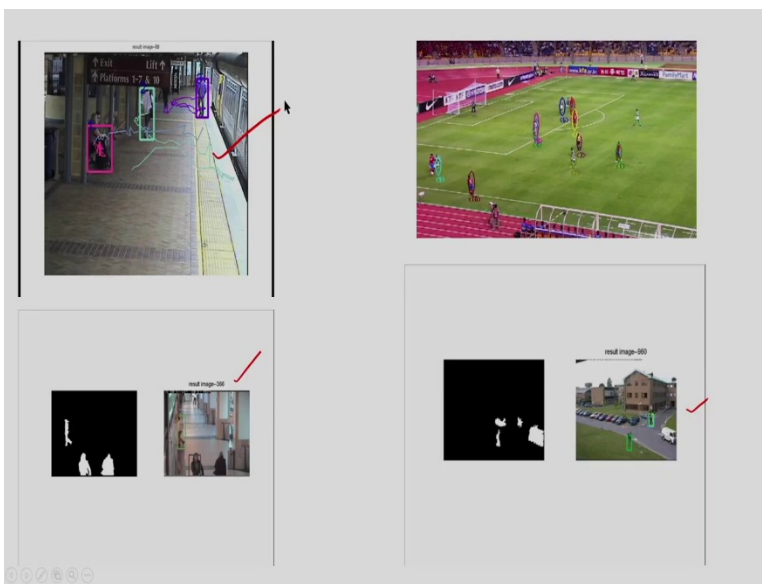

Exit
Lift
Platforms 1-7 & 10

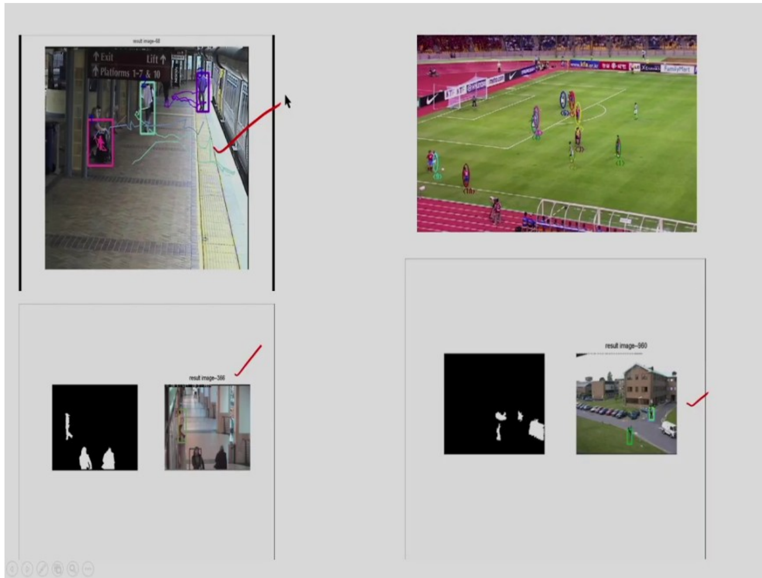

Exit
Lift
Platforms 1-7 & 10

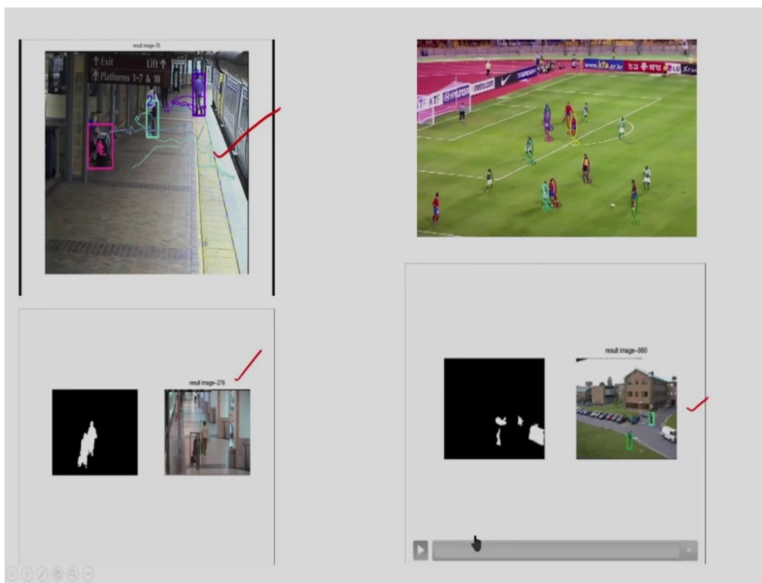


So, in this case you can see the background is cluttered background and also the illumination is also changing. In this case also we are doing the tracking. And in this case you can see, in one case the person is occluded by the car, by the vehicles. And in this case also we are successful in doing the tracking.

(Refer Slide Time: 37:04)

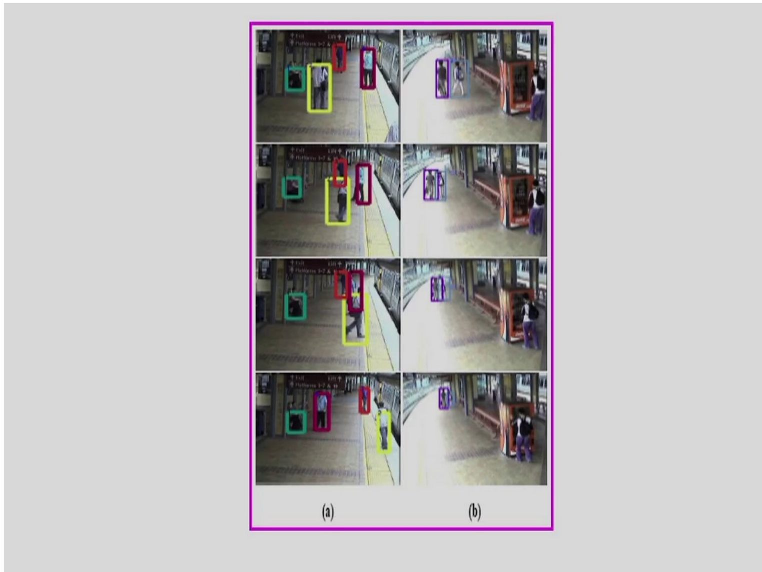






So, this is also another good example of the tracking and here you can see we are finding the trajectory of the movements. So, this person is moving and we are finding the trajectory. So, you can apply the particle filter algorithm for object tracking.

(Refer Slide Time: 37:16)

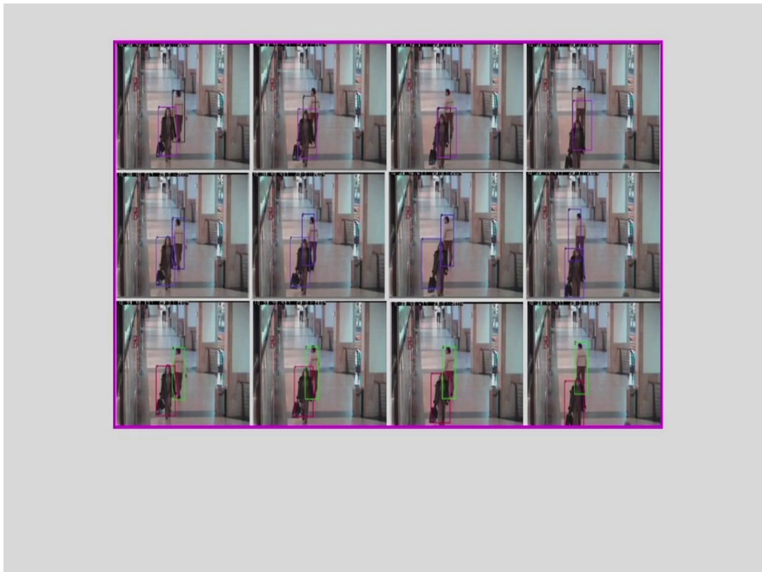


And again I am showing this example of the particle filter based tracking. And sometimes you can see the one person is occluded by another person and in this case also the particle filter can track, particle filter can handle the partial occlusion.

(Refer Slide Time: 37:32)

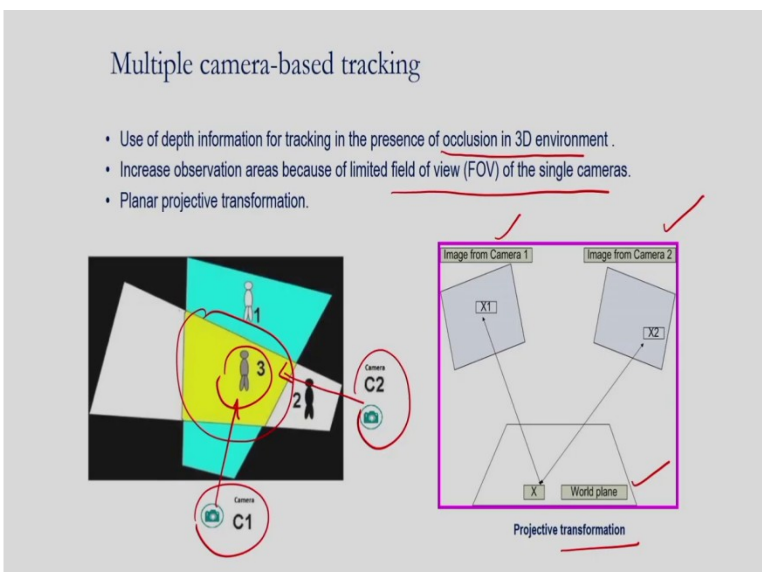
And these are some tracking regions, in some of the cases you can see in this case, in this frame, this persons are occluded by this vehicle and partially this persons are visible. Even then also we can do the tracking. So, the tracking is successful with the help of the particle filter. That means, the particle filter can handle partial occlusion.

(Refer Slide Time: 37:54)



And these are some tracking regions. So, already I have shown the video.

(Refer Slide Time: 38:00)



And also briefly I will explain the concept of the multiple camera based tracking. So, what are the advantages of multiple camera based object tracking? So, one important advantage is that the field of view can be increased. So, in this case you can see, I am considering one camera C1, another camera is C2. So, this person is visible by both the cameras. So, this camera can see this person and this camera can see this person, and you can see the field of view increases because of the use of two cameras.

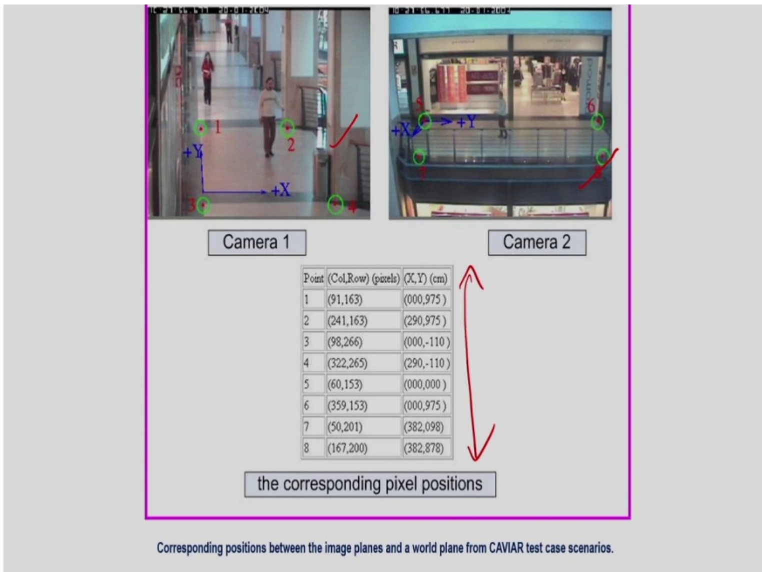
In case of the multiple cameras, we have to find a correspondence between two cameras or multiple cameras. In this example, you can see I am considering the camera 1 and camera 2, and I am finding some transformation to find the correspondence between these two cameras. So, this is the world plane and suppose if I want to track the object, so we have to find a correspondence between camera 1 and camera 2.

So, for this we have to consider some transformation. So, so like maybe some projective transformation we can consider. And another advantage is that, that because the field of view will increase, that means the occlusion can be handled in a multiple camera based system. In case of a single camera based object tracking system, the problem is the occlusion. But in case of the multiple camera based system, since the field of view increases, so the problem of the occlusion can be partially eliminated.

(Refer Slide Time: 39:28)

In this example, you can see one person is completely occluded by a tree. And this is one view of the camera and this is another view of the camera. Maybe in the another view, the person is visible. So, that is why the occlusion problem can be partially resolved by considering multiple camera based systems.

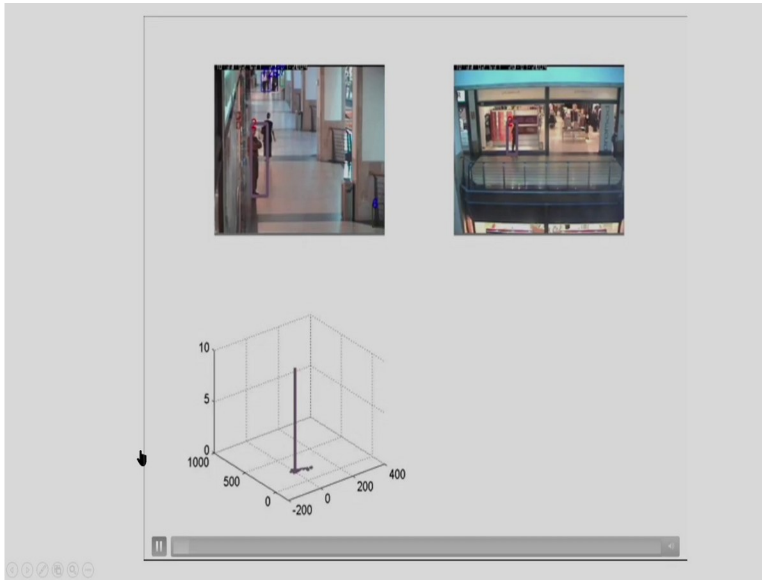
(Refer Slide Time: 39:45)

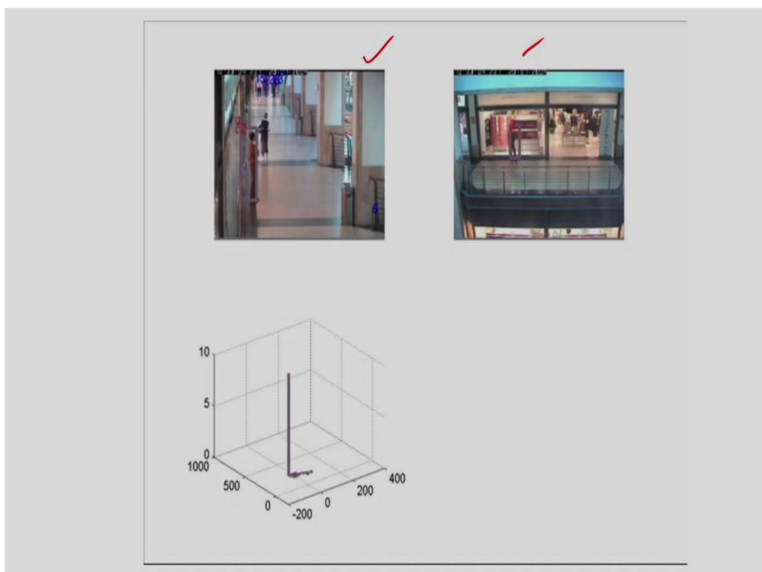


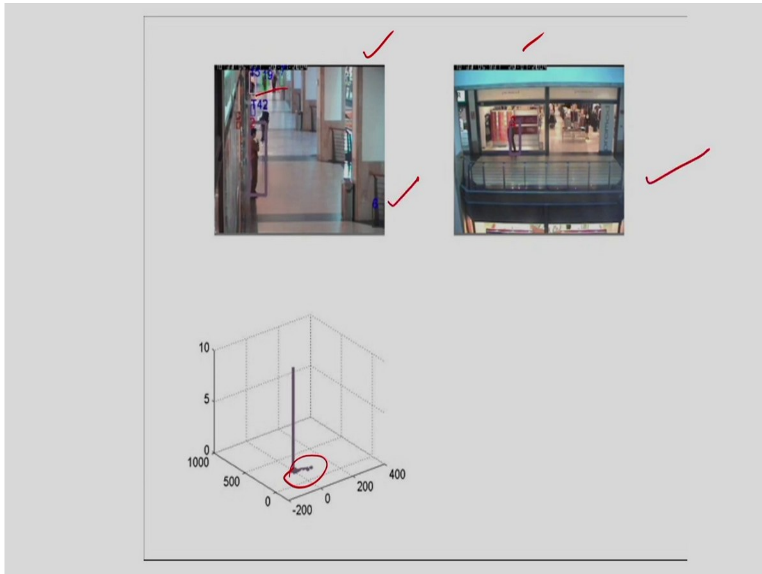
Point	(Col,Row) (pixels)	(X,Y) (cm)
1	(91,163)	(000,975)
2	(241,163)	(290,975)
3	(98,266)	(000,-110)
4	(322,265)	(290,-110)
5	(60,153)	(000,000)
6	(359,153)	(000,975)
7	(50,201)	(382,098)
8	(167,200)	(382,878)

And already I had explained we have to find the correspondence between two cameras or the multiple cameras. Here you can see, and this is the camera 1 view and this is the camera 2 view. And I am finding the correspondence between these two cameras. The corresponding pixel positions. So, based on these corresponding pixel positions, we can do the tracking. So, I am not explaining in detail how to find a correspondence between these two cameras, but concept is that, so we have to find a correspondence between all these cameras in a multiple camera based tracking system.

(Refer Slide Time: 40:18)

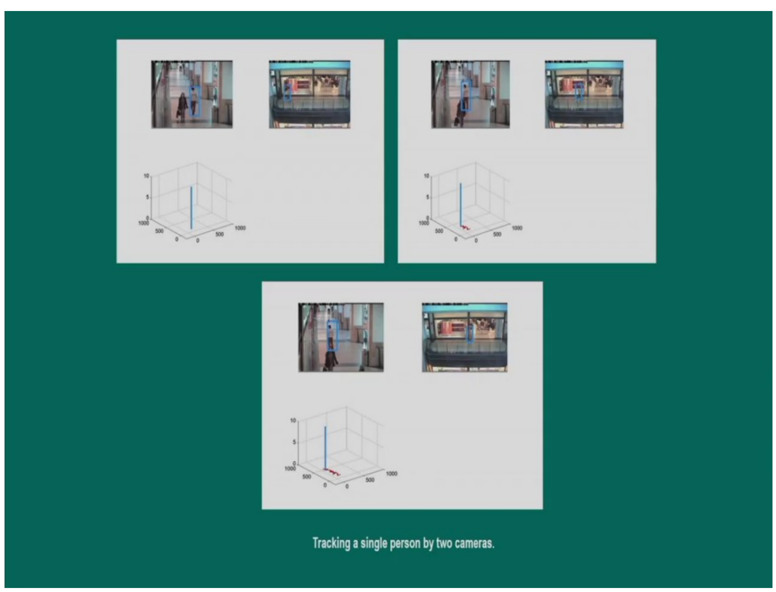






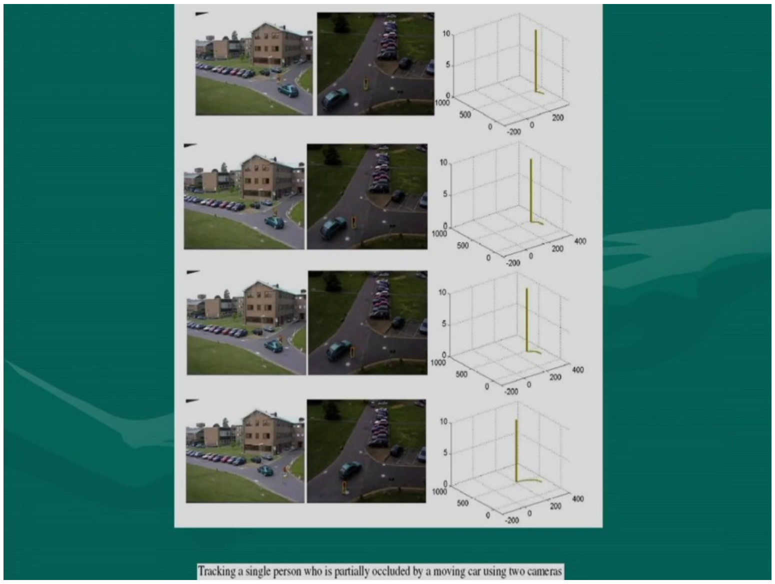
And here I have shown the, one tracking example, the multiple camera base, here I am considering two cameras, camera 1 and camera 2. And I am finding the correspondence between these two cameras. And you can see, just I am also doing the tracking and also I am finding the trajectory of the motion. So, all these persons are tracked. So, this is one view and this is another view of the camera.

(Refer Slide Time: 40:45)



And again, I am showing the results of multiple camera based object tracking.

(Refer Slide Time: 40:50)



And these are some examples of multiple camera based object tracking, tracking single person who is partially occluded by a moving car, using two cameras. Because already I have explained if I use multiple cameras, the problem of occlusion can be partially eliminated, because the field of view increases because of the use of the multiple camera. In this class, I briefly explained the concept of object tracking and the concept of particle filter based object tracking.

For more detail, you have to see the research paper on object tracking and also the research paper on particle filter based object tracking. There are many new techniques of object tracking, the deep learning techniques are also used for object tracking, in a single camera based system and also the multiple camera based system. So for this, you have to see the research, research papers. Let me stop here today. Thank you.