International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Advanced Technique for Real Time Detection and Recognition of Object using Resampling and BPF

Mrs. Sharda R. Chaudhary
Asst. Prof., Dept of Electronics, KITS, Ramtek, Nagpur, Maharastra, India

Mrs. Nutan Y. Suple
Asst. Prof., Dept of Electrical & Communication, KITS, Ramtek, Nagpur, Maharastra, India

Ms. Punam Lambat
Asst. Prof., Department of Computer Technology, KITS, Ramtek, Nagpur, Maharastra, India

ABSTRACT

This paper will explain the bayes filtering system which will explain why should to prefer the particle filters than any other method required for detection of an object in real time videos. It also gives the basic information about the kalman filters, the disadvantages of it and how they are removed in the particle filters. Resampling method related to the pdf of object density for deciding the weight of an object is also included in the paper. The basic steps of particle filters are also well explained. This paper is related to the belief ideas of presence and absence of an object in a particular frame so the false and true conditions can be checked easily than any other method.

Keywords: *particle filter, kalman filter, resampling, Bayesian model, markov model*

I. INTRODUCTION

Particle Filters are always used to find an approximate solution using a complex model rather than an exact solution using a simplified model for tracking the state of a dynamic system modeled by a Bayesian network Similar applications to Kalman Filters, but computationally tractable for large/high dimensional problems[2].

Particle filters are preferred because like Kalman filters, they have a great way to track the state of a dynamic system for which Bayesian model is used. That means that to a model of how the system changes in time, possibly in response to inputs, and a model of what observations should see in particular states, particle filters can be used to track belief state.[3]
The main reason is that for a lot of large or high-dimensional problems, particle filters are tractable Whereas Kalman filters are not. The key idea is that a lot of methods, like Kalman filters, try to make problems more tractable by using a simplified version of full, complex model. Then they can find an exact solution using that simplified model. But sometimes that exact solution is still computationally expensive to calculate, and sometimes a simplified model just isn't good enough. So then we need something like particle filters, which use the full, complex model, but just find an approximate solution instead.[7]

II. BAYES FILTERS

Bayes filters are used for estimating the state of a dynamical system by using concept of prediction and updation to estimate the state of a dynamical system from sensor measurements. Two types of Bayes Filters are Kalman filters and particle filters.[8]

III. METHODOLOGY:

Belief about the current state p(xt | do…t)
Given: ut, zt, perceptual model p(zt | xt), action model p(xt | xt-1, ut-1).

Now we introduce the variables we will be using. X is the state variable, and Xt is the state variable at time t. U is the inputs to your system, z is the observations made by the sensors, and d just refers to inputs and observations together. What the Bayes Filter is trying to find at any point in time is the belief about the current state, which is the probability of xt given all the data we've seen so far.[4]

What we are given is the inputs, the observations, the perceptual model, which is the probability that you'll see a particular observation given that you're in some state at time t, and the action model, which is the probability that you'll end up in state xt at time t, assuming that you started in state xt-1 at time t-1, and input ut-1 to your system.

Particle Filters
(Sequential Monte Carlo)

Figure 1

Above figure represents pdf as a set of samples (particles)
Each particle contains one set of values for the state variables Good for non-Gaussian, multi-model pdfs and is used to find an approximate solution using a complex model (arbitary pdf) rather than an exact solution using a simplified model (Gaussians).

The basic idea of particle filters is that any pdf can be represented as a set of samples (particles). If your pdf looks like the two-humped line in the figure, you can represent that just by drawing a whole lot of samples from it, so that the density of your samples in one area of the state space represents the probability of that region.[8] Each particle has one set of values for the state variables. This method can represent any arbitrary distribution, making it good for non-Gaussian, multi-modal pdfs. Again, the key idea is that you find an approximate representation of a complex model (any arbitrary pdf) rather than an exact representation of a simplified mode (Gaussians).

How to find samples
1. Start the the sample from posterior, p(xt | do…t)
2. (call it p(x) for short) but don't have explicit representation of full pdf to sample.
3. Can sample from prior belief (call it q(x))
4. Sample from prior distribution
5. Update using observations: for each sample, compare p(x) to q(x) and adjust appropriately (find importance weights)

So what you actually want samples of is your posterior, which we will call p(x) for short. But how do we sample from posterior? We don't have an explicit representation of posterior samples to draw points from. But do we know how to sample from prior belief, because some belief from the last time step that we know how to update with motion model. Let's call the prior belief q(x) and do know how to find, for any one x, what the posterior probability is, based on prior belief and observations. So, sample from q(x), and then for each sample that is to be made, update it using what we will call an 'importance weight', based on the observations made.

IV. Particle Filtering Steps

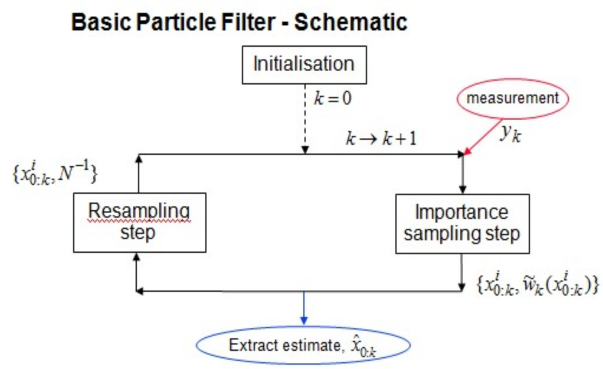

Figure 2

Sample Importance Resampling

Here is a graphical visualization of the importance resampling process.

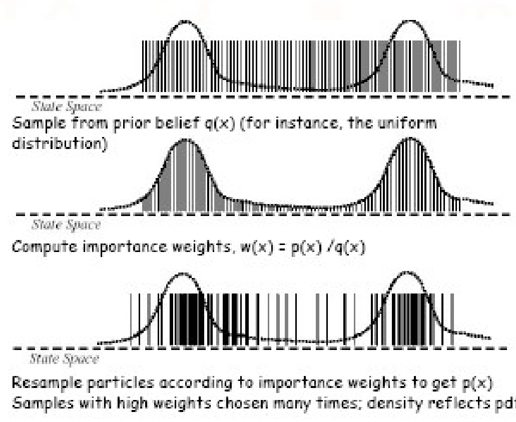

Figure 3

Let's say the posterior we're trying to represent, as before, is the two-humped dotted line.[9] Even if there is no information to start, and our prior is just the uniform distribution, we can still recover the properly sampled pdf of posterior p(x).[10] First sample from prior (the uniform distribution). For each of those samples, find the value of the posterior p(x). So for each sample, assign that sample a weight, w(x), equal to p(x)/q(x). [1] At this point, when the particles are weighted, use highest-weighted (highest-probability) sample as best-guess state, or use the weighted sum of particles to get a mean-equivalent, or use the average of particles within some distance from best particle for a more intelligent best-guess. To represent the pdf properly with samples, though, we want the density of the particles in any segment of the state space to be proportional to the probability of that segment.[5] As we can see in the middle panel, the particles are still spaced evenly from uniform sampling. So in order to adjust the densities properly, we resample the particles. That means we want to keep the total number of particles the same, while increasing the number of particles in the high-probability regions and decreasing the number of particles in low-probability regions. So we draw particles (with replacement) from the set of weighted particles according to their importance weights (probabilities). High-weighted particles can be chosen a lot of times, whereas low-weighted particles are likely not to be chosen at all. The result looks like the third figure, in which the particles go back to being unweighted, and the density of the particles properly represents the pdf.[3]

Discrete Importance Resampling

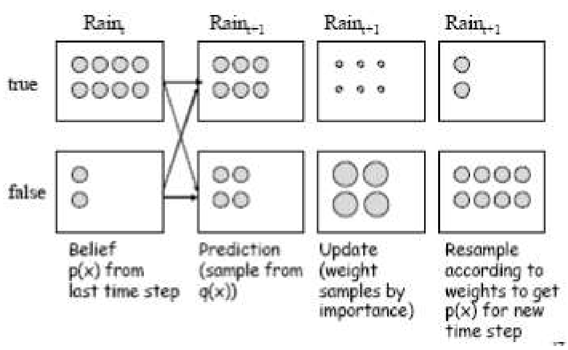

Figure 3

Another way to visualize the importance resampling process is to look at a discrete example.[4]

Let's say we have a dynamic Baye's net with two states: Rain = true or Rain = false.[6] we're trying to figure out whether or not it's raining, but can't see outside because we're in an office with no windows. But let's say that every hour, our boss stops by, and either he brings an umbrella, or he doesn't. If he brings an umbrella, it's likely raining, but maybe not, since some people bring umbrellas for no reason. Likewise, if he doesn't bring an umbrella, it's probably not raining, but it might be. So you have some model about the probability that it's raining, given that you think it was raining an hour ago and our boss brings an umbrella, or doesn't bring an umbrella, and so on. And also have some model about the transition probabilities of rain/not-rain, saying that if it was raining an hour ago, it might have stopped with some probability, and so on. So we start, in the first column of boxes, Raint, with some belief p(x) from the last time step.[2] There are 8 particles in Rain=true and only 2 in Rain=false, meaning that p(rain=true) is 8/(2+8) = 4/5, and p(rain=false) is 2/(2+8) = 1/5. Next, we make a prediction about what the state will be in the next time step based on our transition model, before looking at any observations. This is our prior belief, q(x), and letting particles transition with some probability to each of the two states gives us the new sample set from q(x). Now we have 6 particles in Rain=true, and 4 particles in Rain=false. Then let's say the boss comes in, and he's not carrying an umbrella. Now, we can find the probability of each particle based on our observation, according to our perceptual model. So the Rain=true particles have low probabilities.

Why Resample?

- **If we keep old particles around without resampling:**

Particle depletion Areas with high probability in posterior not represented well Density of particles doesn't represent pdf So, just to make clear why it is necessary to resample the particles: If we just keep old particles around forever without resampling them, what happens is that particles drift around according to motion model (transition probabilities for the next time step), but other than their weights, they are unaffected by observations. Highly unlikely particles will be kept around and transitioned to more unlikely states, and might only have say, one particle in the area of high probability of posterior. So end up with is one particle with a way higher likelihood than any of the other particles, and a whole lot of particles with almost-nil probability.[9] This is what we call 'particle depletion', because in effect have only one particle. And one particle doesn't represent a pdf very well. If we don't have a lot of particles in the areas of pdf with high probability, pdf's can not represent very well.[8] The density of particles should be high in high-probability areas, and low in low-probability areas. And so to resample the particles, so that they continue to represent the pdf accurately and keep track of many high-probability hypotheses, instead of tracking lots of useless, low-probability hypotheses.

CONCLUSIONS AND FUTURE WORK

From the paper it can be concluded that the proposed algorithm can be used for tracking objects in video sequences. Bayes filtering (particle filtering) techniques have been used for estimating the velocity of the moving edge. Particle filter provides an elegant solution to the state-space estimation problem in a highly dimensional non-linear system. Due to the degeneracy of particles, particle filter fails to converge. To overcome these problems, resampling and sample impoverishment techniques have used.

In Figure 4, we display an example of the detection of automobiles and people in a busy street scene.

Figure 4. Example of detecting automobiles and people.

REFERENCES

1. J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamical systems," *Journal of the American Statistical Association*, vol. 93, pp. 1033–1044, 1998.

2. S. Arulampam, S. Maskell, N. Gordon, and T. Clapp,"A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking," *IEEE* Trans. Sig. Proc., vol. 50, pp. 174–188, Feb 2002.

3. A. Doucet, S. J. Godsill, and C. Andrieu, *On Sequential Monte Carlo sampling methods for Bayesian filtering*, vol. 3, Statist. Comput. Commun., July 2002.

4. Y. Nakashima ,N. Babaguchi ,J. Fan, Detecting Intended human objects in Human captured videos. 978-1-4744-7030-3/10 IEEE.

5. Jack Czyz. Object detection in video via particle filtering proceeding of 18th International conference on pattern recognition(ICPR06)0-7695-0/06,IEEE.

6. B.Ristio,SArulampalam,N. Gordon. Beyond kalman filtering :Particle filter for tracking applications .Arten house 2004.

7. K.Tongirala and Kamesh Namuduri, Object tracking in video vis particle filtering ,dept of electrical and computer eng.

8. Young Min Kim object tracking in video sequence CS 229 IEEE final project report.

9. CMPE 264:Image analysis and computer vision ,hai tao,Object tracking and kalman filtering ,dept of computer engg,university of California.

10. M. Jaward ,L.Mihaylava ,Multiple Object tracking using particle filters ,IEEEAC paper #1280,IEEE2006.

11. S.Fazli,H. moradi,H.Bouzari,Particle filter based object tracking with sift and color features ,2009 IEEE second international conference on machine vision.