ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Path Planning and Use of Probabilistic Localization to Keep Track of a Car-Like Robot

Atiya Masood
MS department
Yanbu University and College
Yanbu Al-Saniyah,Saudi Arabia
atiya.masood@yuc.edu.sa, atiyaw@hotmail.com

Abstract: This paper presents a particle filter for tracking a robot car. Particle filter is a way of taking variable of interest about something you are trying to measure, and whittling those guesses down by using measurements from sensors. Robot car also uses its sensors to map the region, uses filtering to estimate its position on the road. Robot car need to have awareness of both static and dynamic parts of the environment. Hence, I propose a model based approach which encompasses both path planning and particle filter for tracking robot car.

Keywords: Particle Filter; Prediction; Monto Carlo localization; Hybride A*, Posterior; Bayes Filter

I. INTRODUCTION

The robot localization is a key problem in mobile robot system [1]. It also has been referred to as “the most fundamental problem of providing a mobile robot with autonomous capabilities” [2]. But the beauty of particle filter is that they provide solution to all mobile robot localization problems that comes in different flavors such as “position tracking”, “global localization”, “kidnap robot problem” and “multi robot localization problem”.

Self driving or robot car is a dream of robotic researcher and it brings to number of benefits to the society, including reducing number of accidents and optimal usages of fuels, safety and comforts. Mobile robot operates in static environments that have maps of their surroundings, whereas robot car need to be aware of both static and dynamic part of environments. This paper describes an approach for tracking a self driving car and path planning of robot car.

For tracking of self driving car numbers of approaches can be used. Different approaches utilized laser finder sometimes in combination with vision [3, 4, 5, and 6]. The vehicle tracking literature almost universally relies on variants of Kaman filters, although particle filters and hybrid approaches have been widely used in other tracking applications [7, 8, and 9].This approach car uses a map of the environment, it knows where the lay marker are and probabilistic localization to keep track of car location. Therefore car could used GPS, Global positioning system, but has enormous error ,some time in order of 5 meter or more meters., which is unsafe for driving, By utilizing particle filter with multimodal *pdf car can* do same with about 10 cm error, which mean it can really understand where to stay in the lane just by knowing where the lane is in advance and using localization techniques. Mobile robot has a map of the environment and the environment itself is static. This is clearly not true for a self-driving car. Obstacles will be detected at all times that the robot doesn't know about. For this reason the approach models both path planning and particle filter properties of the tracked vehicles

II. REPRESENTATION

A. Particle Filter:

Particle Filter is a hypothesis tracker that approximates the filtered posterior distribution by a set of weighted particles[10]. It weights particles based on a likelihood score and then propagates these particles according to a motion model. The Particle filter operates in two phases’ prediction and update. In prediction stage each particle is modified according to the existing model, including the addition of random noise in order to simulate the effect of noise on the variable of interest .Then each particle weight is revaluated based on the latest sensory information available, it is called update stage of particle filter [11].

From a mathematical perspective, particle filters estimate the posterior over unobservable state variables from sensor measurements. In the context of the present paper, *state* refers to the position of the robot car (location and orientation) relative to its environment, along with the number of obstacles in the robot’s proximity. For the sake of the general discussion of particle filters, the total of all those state variables will be denoted by ***x***.

In particular, let the state at time ***t*** denoted by x_t . Particle filters address situations in which this state is not directly observable. Instead, the robot must rely on sensor measurements and information about the controls it executes to infer the posterior distribution over *x* .Let z_t denote the sensor measurement acquired at time *t* and u_t *denote the* control at time *t.* Thus at time *t* two type of information relevant to the state x_t are available to the robot.

$$z^t := \{z_1, \ldots\ldots zt\} \qquad (1)$$

$$u^t = \{u_1, \ldots\ldots ut\} \qquad (2)$$

The goal of particle filtering is to estimate the posterior probability over the state variable x at time t

$$p(x_t|z^t,u^t) \qquad (3)$$

This posterior is calculated recursively

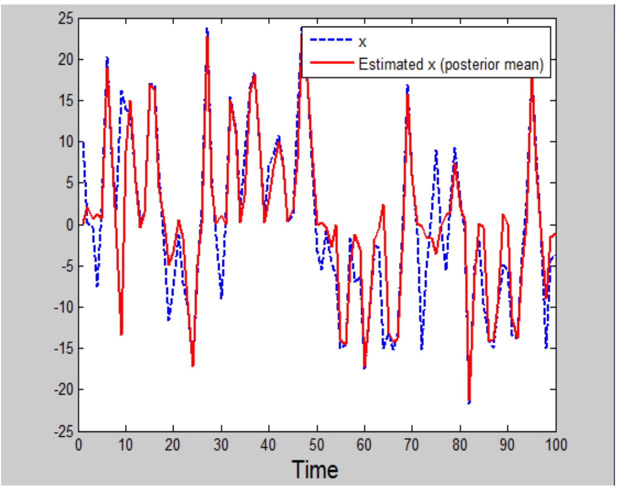
$$p(x_t|z^t,u^t)=\eta\ p(z_t|x_t)\int p(xt\,|ut\ ,xt-1)\ p(xt\ -1|\ zt-1, ut-1)dxt-1 \quad (4)$$

Here η is a constant normalizer. The conditional probability distribution $p(z_t \mid x_t)$ is a *measurement model* Similarly, $p(x_t\,|u_t\ ,\ x_{t-1})$ is a *motion model.* The recursive update equation (4) is equivalent to the well-known Bayes-filters.The key idea of the particle filter is to approximate this posterior by set of hypothesized states called *particles*. Which are distributed according to p $(x_t|z^t,u^t)$. Put mathematically $p(x_t|z^t,u^t)$ is represented by set of particles.

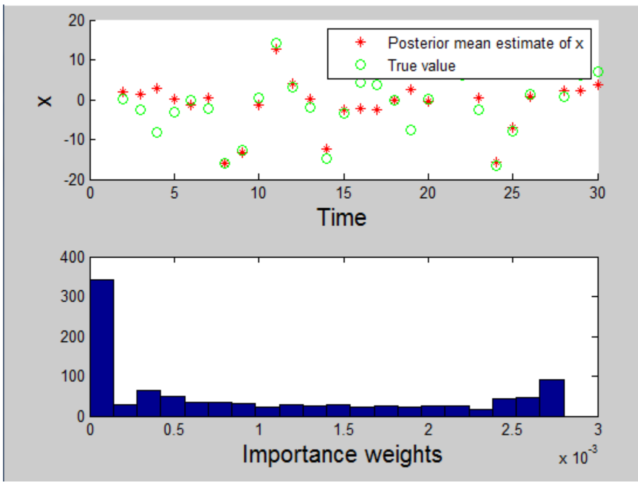
$$Xt:=\{x_t^{[i]}\}\ i=1,\ldots.N \qquad (5)$$

Where N is a size of particles. It is well-known that such a set of particles x_t can be obtained via the following sampling procedure, which is directly derived from the recursive update equation (3).

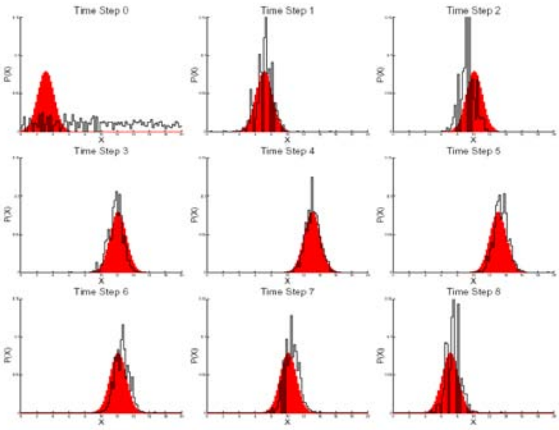
$x_t=\emptyset$
for i=1 to N do
take $x_{t-1}^{[i]}$ *from* xt_{-1}
draw$x_t^{[i]} \sim p(x_t|u_t\ ,x_{t-1}^{[i]})$
calculate (non –normalized) weight
$w_t^{[i]}=p(z_t|x_t^{[i]})$
endfor
for i=1 to N do
draw k with probability $w_t^{[k]}/\sum_t^N wt[i]$
add $x_t^{[k]}$ *to* x
endfor



(a)



(b)

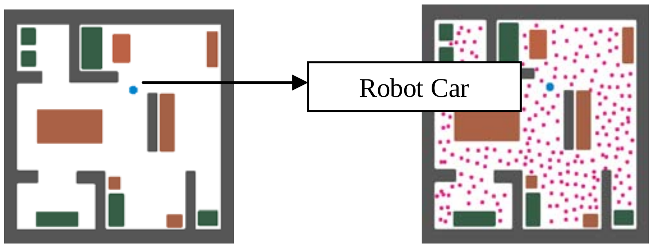


(c)

Figure.1.(a) Estimate particles and posterior mean (b) Imporatnt weights of partcles and posterior mean estimate of x (c) Moving Gaussian + uniform, N=100 particles Prediction

B. Prediction

In order to predict the probability distribution of pose of the robot car after a motion need to have a model of the effect of noise on the resulting pose. In the present case, the robot car has a map of the environment, and some sensor that measures how far it is away from an obstruction. Now the robot car does not know its x, y coordinate.



(a) (b)

Fig.ure 2 . (a) Actual position of robot car (b) Each particle has three diminsion vectors .

A single point will carry three values (X,Y,ϴ) for X position, Y position, and ϴ heading .Robot car can drive forward along the heading, or rotate its heading. Each particle has three dimension vectors.

$$X'=X+V\Delta t\cos\theta \qquad (6)$$

$$Y'=Y+V\Delta t\sin\theta \qquad (7)$$

$$\theta' = \theta + \omega t \qquad (8)$$

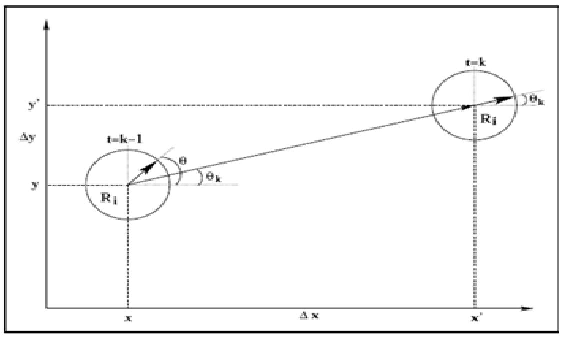


Fig.ure 3 . Arbitrary motion of robot car Ri. At time t [x,y,θ], after the motion the pose is [x',y' Θ']

Exact outcome cannot be predicted by MontoCarlo localization or particle filter, for this reason noise add to velocity v and ω. Robot car has lots of particles in it environment and each particle has slightly different x,y coordinate and heading outcome. These particles comprise together for estimation after the motion command. If single particles drawn multiple times, give a set of particles shown in the Figure 4. Vary v and ω with little bit of noise give slightly different prediction and result get particle as in Figure 4

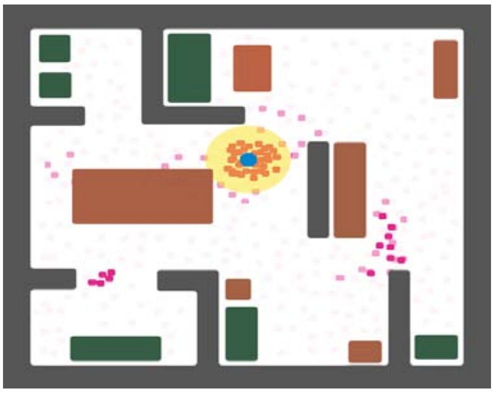


Figure.4 . Single particles drawn multiple times Robot Path Planning

C. Robot Path Planning

Robot path planning and motion planning is a rich field itself but difference between planning algorithm and robot path planning is that robot world is continuous .

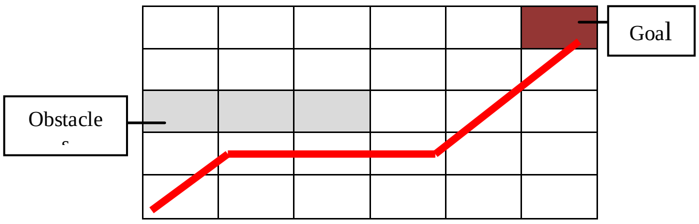


Figure.5. A* use in a discrete world

A car cannot really follow these discrete choices. Numbers of very sharp turn are irreconcilable with the motion of the car. So fundamental problem of A* is discrete, whereas the robotic world is continuous. For solving this problem with A* has to do with the state transition function and this called hybrid A*, it memorize a next x', y'and Θ'.

Actual planning technique uses dynamic programming and understand how far thing are away? It also rollout to avoid local obstacles. These local rollouts are continuous trajectories. They are variant by discrete decision, like weather to change the lane, and by various small discrete nudges around obstacles. This is how it can avoid obstacles. And in rolling out to a certain horizon, By using a dynamic programming actual traffic situation can be calculated and also the cost of the critical path. This method has really navigated very complicated situation with self driving car.

III. DISCUSSION

This paper presented the particle filter, the recursive process that a robot can use to localize itself. Partially observable and stochastic environment need to simultaneously plan and execute. So the car has to plan and measure weight by using sensors to determine the location and perhaps the identity of all relevant objects, and localize it. GPS readings are not reliable enough for accurate positioning.

Robot car uses its sensors to map the region, uses filtering to estimate its position on the road given the observations of car data and sensors, and then uses planning (hybrid A*, etc.) to adjust the current plan.

IV. ACKNOWLEDGMENT

I gratefully acknowledge the Yanbu University and College for every support of this research.

V. REFERENCES

[1] www.cfar.umd.edu|~fer|cmsc828|classes|fox.mcmc

[2] book.pdf**robot**s.stanford.edu|papers|thrun.landmark.pdf

[3] L. Zhao and C. Thorpe. Qualitative and quantitative car tracking from a range image sequence. In Computer Vision and Pattern Recognition, 1998.

[4] D. Streller, K. Furstenberg, and K. Dietmayer. Vehicle and object dels for robust tracking in traffic scenes using laser range images. In Intelligent Transportation Systems, 2002.

[5] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research, Sep 2007; vol. 26: pp. 889-916, 2007

[6] S. Wender and K. Dietmayer. 3d vehicle detection using a laser scanner and a video camera. In 6th European Congress on ITS in Europe,Aalborg, Denmark, 2007

[7] S. S¨arkk¨a, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. Inf. Fusion, 8(1), 2007.

[8] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In ICRA, 2001.

[9] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. IEEE AE Systems Magazine, 2004.

[10] Eugenia Koblents, Joaqu´ýn M´ýguez . A Poppulation Carlo Method For Bayesian Inference and its application to stochastic Kinetic Models

[11] A particle filter tutorial for Mobile robot localization TR-CIM-0402, Ioannis M.Rekletities.[11] D. MacKay, Learning in Graphical Models, chapter Introduction to Monte Carlo Methods, pp. 175–204, Kluwer Academic Publishers,1998.