AA274A: Principles of Robot Autonomy I
Course Notes

Previous chapters introduced several algorithms for robot localization and state estimation that are based on a probabilistic framework. In particular, the Bayes filter was first introduced as a fundamental approach to the problem, which uses a probabilistic state transition model and a measurement model to recursively update a belief distribution over possible states. A set tractable implementations of the Bayes filter that model the belief distribution in a *parametric* way, for example using Gaussian distributions, was then presented (in particular the Kalman and extended Kalman filters). These filters leverage the structure of the parametric belief distribution to provide a computationally efficient approach to dealing with continuous state spaces (which have an infinite number of states). For example the Gaussian distribution represents a continuous distribution through a *finite* set of parameters: the mean and covariance. However there are also other implementations of Bayes filter that can be efficiently used in continuous state spaces that are *non-parametric.*

16 Nonparametric Filters

In contrast to parametric filters, *non-parametric* filters do not make assumptions on the structure of the belief distribution. This can be a desirable property for applications in robotics where rigid structures in the belief distribution may result in poor performance. A classic example is that the Gaussian distributions used in the Kalman filter and EKF are unimodal, which cannot express the possibility that two distinct "high probability" states might exist at the same time. Non-parametric filters on the other hand generally represent the belief distribution in an unstructured way, for example through a finite number of samples drawn from the distribution, which allows for more expressive distributions. This chapter introduces two main approaches for non-parametric filtering: the *histogram filter* and the *particle filter.*

16.1 Histogram Filter

The histogram filter is essentially a modification of the discrete Bayes filter presented earlier to work in continuous state spaces. In particular, the continuous state space is decomposed into a finite number of regions and the belief is represented over the discretized space by collecting the finite number of probabilities of the state being in each discretized region.

In particular for the random state variable X_t, the continuous state space $\mathrm{dom}(X_t)$ is decomposed into a finite set of regions (often called *bins* in the context of histogram filters):

$$\mathrm{dom}(X_t) = \boldsymbol{x}_{1,t} \cup \boldsymbol{x}_{2,t} \cup ... \cup \boldsymbol{x}_{K,t}, \tag{1}$$

where $\boldsymbol{x}_{k,t}$ is the k-th "bin". For example, if the one-dimensional random variable X could take on values in the interval $[a, b]$ then one possible decomposition would be to split the interval into a finite number of sub-intervals with equal width. The belief distribution is then defined in non-parametric way by simply specifying a probability $p_{k,t}$ to each bin $\boldsymbol{x}_{k,t}$. A probability density function can then be defined in a piecewise manner:

$$p(\boldsymbol{x}_t) = \frac{p_{k,t}}{|\boldsymbol{x}_{k,t}|}, \quad \boldsymbol{x}_t \in \boldsymbol{x}_{k,t}, \tag{2}$$

where $|\boldsymbol{x}_{k,t}|$ denotes the "area" or "volume" of the bin. This definition implies that the probability that the random variable X_t takes on *any* value in the bin $\boldsymbol{x}_{k,t}$ is equal to $p_{k,t}$.

The prediction and measurement update steps of the Bayes filter are then accomplished by also discretizing the state transition and measurement models by computing a representative "mean" state for each bin:

$$\hat{\boldsymbol{x}}_{k,t} = |\boldsymbol{x}_{k,t}|^{-1} \int_{\boldsymbol{x}_{k,t}} x_t d\boldsymbol{x}_t. \tag{3}$$

The state transition model $p(\boldsymbol{x}_{k,t} \mid \boldsymbol{u}_t, \boldsymbol{x}_{i,t-1})$ that defines the probability of transitioning from one bin to another is then approximated in terms of the mean bin states by:

$$p(\boldsymbol{x}_{k,t} \mid \boldsymbol{u}_t, \boldsymbol{x}_{i,t-1}) \approx \eta |\boldsymbol{x}_{k,t}| p(\hat{\boldsymbol{x}}_{k,t} \mid \boldsymbol{u}_t, \hat{\boldsymbol{x}}_{i,t-1}), \tag{4}$$

where $p(\hat{\boldsymbol{x}}_{k,t} \mid u_t, \hat{\boldsymbol{x}}_{i,t-1})$ is the original (non-discretized) state transition model evaluated at the mean bin states $\hat{\boldsymbol{x}}$ and η is a normalization constant[1].

The discretization of the measurement model is accomplished in a similar manner, with the discretized model given by:

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_{k,t}) \approx p(\boldsymbol{z}_t \mid \hat{\boldsymbol{x}}_{k,t}). \tag{5}$$

In other words, the measurement probability associated with a bin is approximated by the measurement probability associated with the mean bin state $\hat{\boldsymbol{x}}_{k,t}$.

After the discretization has been performed, the discrete Bayes filter algorithm from before can be directly applied by iterating over each bin and updating the probability $p_{k,t}$.

16.2 Particle Filter

The particle filter is another non-parametric filter that provides a computationally tractable implementation of the Bayes filter for continuous state spaces. This filter represents the belief distribution by a finite set of random samples called particles, which are denoted by:

$$\mathcal{X}_t \coloneqq \{\boldsymbol{x}_t^{[1]}, \boldsymbol{x}_t^{[2]}, ..., \boldsymbol{x}_t^{[M]}\}. \tag{6}$$

[1] In the case that the bin areas $|\boldsymbol{x}_{k,t}|$ are equal, these terms can be absorbed into the normalization constant.

Each particle $x_t^{[m]}$ represents a hypothesis about the true state $\boldsymbol{x}_t$, and therefore regions of the state space with more particles correspond to regions of high probability. Ideally, the particles are distributed according to the current belief:

$$\boldsymbol{x}_t^{[m]} \sim p(\boldsymbol{x}_t \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) = bel(\boldsymbol{x}_t), \tag{7}$$

but theoretically this only occurs as $M \to \infty$. Instead the set of particles approximately represents the belief distribution, and in practice around $M \approx 1000$ samples tends to be sufficient (but of course this depends on the application).

The particle filter updates the belief (via a prediction and measurement correction step) by manipulating the prior set of particles $\mathcal{X}_{t-1}$ to yield a new set of particles $\mathcal{X}_t$. The prediction step is implemented by considering each particle $\boldsymbol{x}_{t-1}^{[m]}$ in the prior set $\mathcal{X}_{t-1}$ and sampling from the state transition model a new "predicted" sample $\bar{\boldsymbol{x}}_t^{[m]} \sim p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{x}_{t-1}^{[m]})$. An importance factor $w_t^{[m]}$ is then defined for the predicted sample $\bar{\boldsymbol{x}}_t^{[m]}$ based on how well the observed measurement matches the prediction. Specifically, the importance factor is computed as $w_t^{[m]} = p(\boldsymbol{z}_t \mid \bar{\boldsymbol{x}}_t^{[m]})$. The predicted particles $\bar{\boldsymbol{x}}_t^{[m]}$ and their associated weights $w_t^{[m]}$ can then be collected in a new particle set $\bar{\mathcal{X}}_t$, which represents the predicted belief distribution $\overline{bel}(\boldsymbol{x}_t)$. The correction step is then accomplished by simply resampling (with replacement) a new set of M particles from the predicted set $\bar{\mathcal{X}}_t$ with a probability proportional to the weights $w_t^{[m]}$. This procedure performs the measurement correction by giving preference in the new sample set to those predicted particles that showed higher correlation to the measurement $\boldsymbol{z}_t$. The resampled points are then collected in a new set $\mathcal{X}_t$ that defines the posterior belief distribution. This algorithm is also outlined in Algorithm 1 and a few iterations of the algorithm for a simple robot localization problem are shown in Figure 1.

```
Data: $\mathcal{X}_{t-1}, \boldsymbol{u}_t, \boldsymbol{z}_t$
Result: $\mathcal{X}_t$
$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $m = 1$ to $M$ do
    Sample $\bar{\boldsymbol{x}}_t^{[m]} \sim p(\boldsymbol{x}_t \mid \boldsymbol{u}_t, \boldsymbol{x}_{t-1}^{[m]})$
    $w_t^{[m]} = p(\boldsymbol{z}_t \mid \bar{\boldsymbol{x}}_t^{[m]})$
    $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t \cup \big(\bar{\boldsymbol{x}}_t^{[m]}, w_t^{[m]}\big)$
end
for $m = 1$ to $M$ do
    Draw $i$ with probability $\propto w_t^{[i]}$ Add $\bar{\boldsymbol{x}}_t^{[i]}$ to $\mathcal{X}_t$
end
return $\mathcal{X}_t$
```

Algorithm 1: Particle Filter Algorithm

Note that the concept of resampling in the correction step can be quite important for reasons beyond just updating the belief for the measurement correction. In particular, without the resampling step over time some of the particles would drift to regions of low probability and there would be fewer particles to represent the regions of high probability. The resampling step can therefore be viewed as a probabilistic implementation of the Darwinian idea of survival of the fittest: it refocuses the particle set to regions in state space with high posterior probability. This helps from a computational efficiency standpoint because it reduces the number of particles that are needed by focusing them on the regions of the state space that matter (i.e. regions of high probability).

References

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* MIT press, 2005.

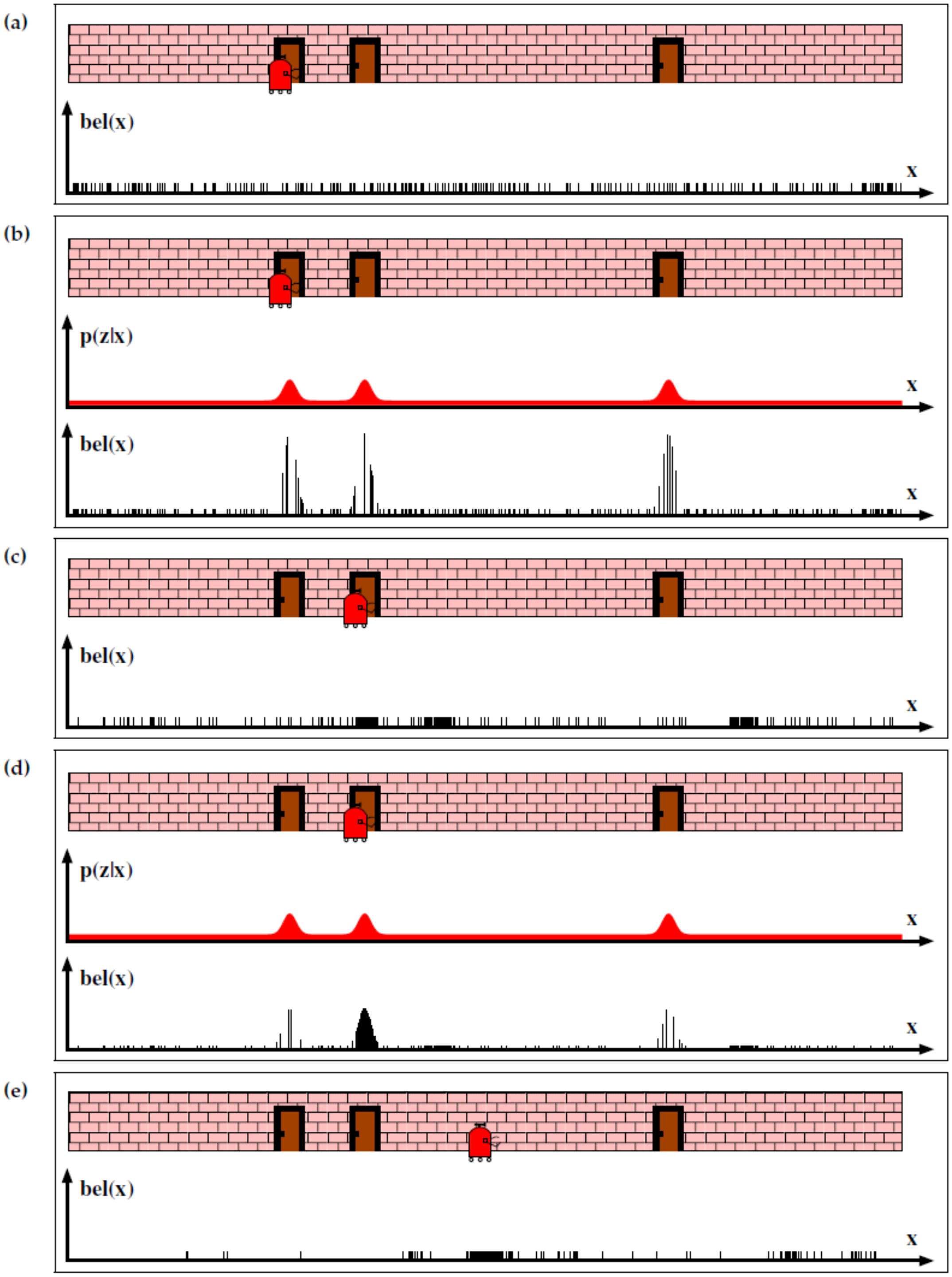

Figure 1: Particle filter used for robot localization from [TBF05]: (a) Initial particles sampled uniformly over entire state space, (b) same set of particles from (a) after importance weighting with initial sensor measurement, (c) resampled particles from weighted distribution after motion, (d) importance weighting of new particle set with new sensor measurement, and (e) resampled particle set after further motion.