approximate the measurement function h by Taylor expansion:

(13.15)
$$h(m_{c_t}, x_t^{[k]}) \approx \underbrace{h(\mu_{c_t,t-1}^{[k]}, x_t^{[k]})}_{=:\,\hat{z}_t^{[k]}} + \underbrace{h'(x_t^{[k]}, \mu_{c_t,t-1}^{[k]})}_{=:\,H_t^{[k]}}(m_{c_t} - \mu_{c_t,t-1}^{[k]})$$
$$= \hat{z}_t^{[k]} + H_t^{[k]}\,(m_{c_t} - \mu_{c_t,t-1}^{[k]})$$

Here the derivative h' is taken with respect to the feature coordinates m_{c_t}. This linear approximation is tangent to h at $x_t^{[k]}$ and $\mu_{c_t,t-1}^{[k]}$. Under this approximation, the posterior for the location of feature c_t is indeed Gaussian. The new mean and covariance are obtained using the standard EKF measurement update:

(13.16)
$$K_t^{[k]} = \Sigma_{c_t,t-1}^{[k]} H_t^{[k]T} (H_t^{[k]} \Sigma_{c_t,t-1}^{[k]} H_t^{[k]T} + Q_t)^{-1}$$

(13.17)
$$\mu_{c_t,t}^{[k]} = \mu_{c_t,t-1}^{[k]} + K_t^{[k]}(z_t - \hat{z}_t^{[k]})$$

(13.18)
$$\Sigma_{c_t,t}^{[k]} = (I - K_t^{[k]} H_t^{[k]})\Sigma_{c_t,t-1}^{[k]}$$

Steps 1 and 2 are repeated M times, resulting in a temporary set of M particles.

3. **Resampling.** In a final step, FastSLAM resamples this set of particles. We already encountered resampling in a number of algorithms. FastSLAM draws from its temporary set M particles (with replacement) according to a yet-to-be-defined importance weight. The resulting set of M particles then forms the new and final particle set, Y_t. The necessity to resample arises from the fact that the particles in the temporary set are not distributed according to the desired posterior: Step 1 generates poses x_t only in accordance with the most recent control u_t, paying no attention to the measurement z_t. As the reader should know well by now, resampling is the common technique in particle filtering to correct for such mismatches.

   This situation is once again illustrated in Figure 13.5, for a 1-D example. Here the dashed line symbolizes the *proposal distribution*, which is the distribution at which particles are generated, and the solid line is the target distribution. In FastSLAM, the proposal distribution does not depend on z_t, but the target distribution does. By weighing particles as shown in the bottom of this figure, and resampling according to those weights, the resulting particle set indeed approximates the target distribution.

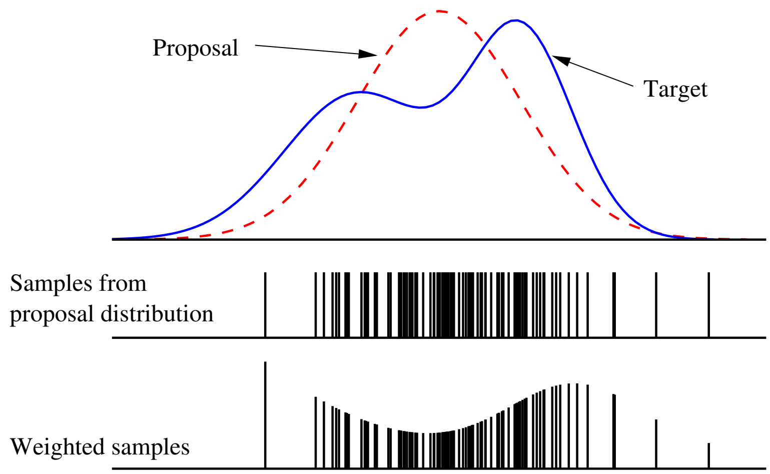

Figure 13.5 Samples cannot be drawn conveniently from the target distribution (shown as a solid line). Instead, the importance sampler draws samples from the proposal distribution (dashed line), which has a simpler form. Below, samples drawn from the proposal distribution are drawn with lengths proportional to their importance weights.

To determine the importance factor, it will prove useful to calculate the actual proposal distribution of the path particles in the temporary set. Under the assumption that the set of path particles in Y_{t-1} is distributed according to $p(x_{1:t-1} \mid z_{1:t-1}, u_{1:t-1}, c_{1:t-1})$ (which is an asymptotically correct approximation), path particles in the temporary set are distributed according to:

(13.19) $$p(x_{1:t}^{[k]} \mid z_{1:t-1}, u_{1:t}, c_{1:t-1}) = p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)\, p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1}, c_{1:t-1})$$

The factor $p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)$ is the sampling distribution used in Equation (13.12).

The *target distribution* takes into account the measurement at time z_t, along with the correspondence c_t:

(13.20) $$p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t}, c_{1:t})$$

The resampling process accounts for the difference of the target and the proposal distribution. As usual, the *importance factor* for resampling is given by the quotient of the target and the proposal distribution:

$$w_t^{[k]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t}, c_{1:t})}{p(x_{1:t}^{[k]} \mid z_{1:t-1}, u_{1:t}, c_{1:t-1})} = \eta\, p(z_t \mid x_t^{[k]}, c_t) \tag{13.21}$$

The last transformation is a direct consequence of the following transformation of the enumerator in (13.21):

$$\begin{aligned} & p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t}, c_{1:t}) \\ &= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}, c_{1:t})\, p(x_{1:t}^{[k]} \mid z_{1:t-1}, u_{1:t}, c_{1:t}) \\ &= \eta\, p(z_t \mid x_t^{[k]}, c_t)\, p(x_{1:t}^{[k]} \mid z_{1:t-1}, u_{1:t}, c_{1:t-1}) \end{aligned} \tag{13.22}$$

To calculate the probability $p(z_t \mid x_t^{[k]}, c_t)$ in (13.21), it will be necessary to transform it further. In particular, this probability is equivalent to the following integration, where we once again omit variables irrelevant to the prediction of sensor measurements:

$$\begin{aligned} w_t^{[k]} &= \eta \int p(z_t \mid m_{c_t}, x_t^{[k]}, c_t)\, p(m_{c_t} \mid x_t^{[k]}, c_t)\, dm_{c_t} \\ &= \eta \int p(z_t \mid m_{c_t}, x_t^{[k]}, c_t)\, \underbrace{p(m_{c_t} \mid x_{1:t-1}^{[k]}, z_{1:t-1}, c_{1:t-1})}_{\sim \mathcal{N}(\mu_{c_t,t-1}^{[k]}, \Sigma_{c_t,t-1}^{[k]})}\, dm_{c_t} \end{aligned} \tag{13.23}$$

Here $\mathcal{N}(x; \mu, \Sigma)$ denotes a Gaussian distribution over the variable x with mean μ and covariance Σ.

The integration in (13.23) involves the estimate of the observed feature location at time t and the measurement model. To calculate (13.23) in closed form, FastSLAM employs the very same linear approximation used in the measurement update in Step 2. In particular, the importance factor is given by

$$w_t^{[k]} \approx \eta\, |2\pi Q_t^{[k]}|^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(z_t - \hat{z}_t^{[k]}) Q_t^{[k]-1} (z_t - \hat{z}_t^{[k]})\right\} \tag{13.24}$$

with the covariance

$$Q_t^{[k]} = H_t^{[k]T}\, \Sigma_{n,t-1}^{[k]}\, H_t^{[k]} + Q_t \tag{13.25}$$

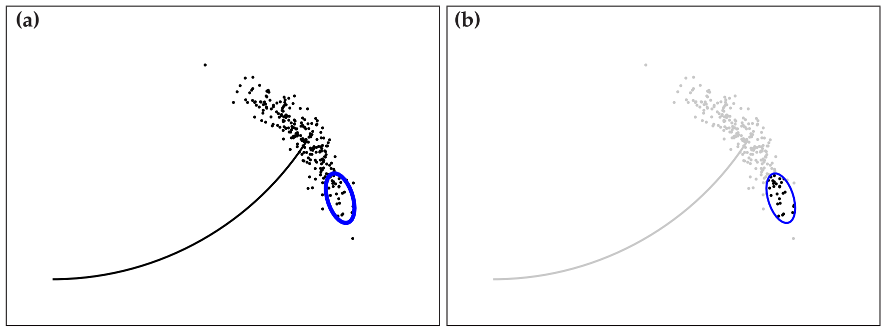

Figure 13.6 Mismatch between proposal and posterior distributions: (a) illustrates the forward samples generated by FastSLAM 1.0, and the posterior induced by the measurement (ellipse). Diagram (b) shows the sample set after the resampling step.

This expression is the probability of the actual measurement z_t under the Gaussian. It results from the convolution of the distributions in (13.23), exploiting our linear approximation of h. The resulting importance weights are used to draw with replacement M new samples from the temporary sample set. Through this resampling process, particles survive in proportion of their measurement probability.

These three steps together constitute the update rule of the FastSLAM 1.0 algorithm for SLAM problems with known data association. We note that the execution time of the update does not depend on the total path length t. In fact, only the most recent pose $x_{t-1}^{[k]}$ is used in the process of generating a new particle at time t. Consequently, past poses can safely be discarded. This has the pleasing consequence that neither the time requirements nor the memory requirements of FastSLAM depend on the total number of time steps spent on data acquisition.

A summary of the FastSLAM 1.0 algorithm with known data association is provided in Table 13.1. For simplicity, this implementation assumes that only a single feature is measured at each point in time. This algorithm implements the various update steps in a straightforward manner. Its implementation is relatively straightforward; in fact, FastSLAM 1.0 happens to be one of the easiest SLAM algorithms to implement!

Algorithm FastSLAM 1.0_known_correspondence(z_t, c_t, u_t, Y_{t-1}):

    for $k = 1$ *to* M *do* — *// loop over all particles*
        retrieve $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots, \left\langle \mu_{N,t-1}^{[k]}, \Sigma_{N,t-1}^{[k]} \right\rangle \right\rangle$ *from* Y_{t-1}
        $x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ — *// sample pose*
        $j = c_t$ — *// observed feature*
        if feature j *never seen before*
            $\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ — *// initialize mean*
            $H = h'(x_t^{[k]}, \mu_{j,t}^{[k]})$ — *// calculate Jacobian*
            $\Sigma_{j,t}^{[k]} = H^{-1}\, Q_t\, (H^{-1})^T$ — *// initialize covariance*
            $w^{[k]} = p_0$ — *// default importance weight*
        else
            $\hat{z} = h(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ — *// measurement prediction*
            $H = h'(x_t^{[k]}, \mu_{j,t-1}^{[k]})$ — *// calculate Jacobian*
            $Q = H\, \Sigma_{j,t-1}^{[k]}\, H^T + Q_t$ — *// measurement covariance*
            $K = \Sigma_{j,t-1}^{[k]}\, H^T\, Q^{-1}$ — *// calculate Kalman gain*
            $\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z})$ — *// update mean*
            $\Sigma_{j,t}^{[k]} = (I - K\, H)\Sigma_{j,t-1}^{[k]}$ — *// update covariance*
            $w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}_n)^T Q^{-1} (z_t - \hat{z}_n)\right\}$ — *// importance factor*
        endif
        for all other features $j' \neq j$ *do* — *// unobserved features*
            $\mu_{j',t}^{[k]} = \mu_{j',t-1}^{[k]}$ — *// leave unchanged*
            $\Sigma_{j',t}^{[k]} = \Sigma_{j',t-1}^{[k]}$
        endfor
    endfor
    $Y_t = \emptyset$ — *// initialize new particle set*
    do M *times* — *// resample* M *particles*
        draw random k *with probability* $\propto w^{[k]}$ — *// resample*
        add $\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \ldots, \left\langle \mu_N^{[k]}, \Sigma_N^{[k]} \right\rangle \right\rangle$ *to* Y_t
    endfor
    return Y_t

Table 13.1 FastSLAM 1.0 with known correspondence.

13.4 Improving the Proposal Distribution

FASTSLAM 2.0

FastSLAM 2.0 is largely equivalent to FastSLAM 1.0, with one important exception: Its proposal distribution takes the measurement z_t into account, when sampling the pose x_t. By doing so it can overcome a key limitation of FastSLAM 1.0.

On the surface, the difference looks rather marginal: The reader may recall that FastSLAM 1.0 samples poses based on the control u_t only, and then uses the measurement z_t to calculate importance weights. This is problematic when the accuracy of control is low relative to the accuracy of the robot's sensors. Such a situation is illustrated in Figure 13.6. Here the proposal generates a large spectrum of samples shown in Figure 13.6a, but only a small subset of these samples have high likelihood, as indicated by the ellipsoid. After resampling, only particles within the ellipsoid "survive" with reasonably high likelihood. FastSLAM 2.0 avoids this problem by sampling poses based on the measurement z_t in addition to the control u_t. Thus, as a result, FastSLAM 2.0 is more efficient than FastSLAM 1.0. On the downside, FastSLAM 2.0 is more difficult to implement than FastSLAM 1.0, and its mathematical derivation is more involved.

13.4.1 Extending the Path Posterior by Sampling a New Pose

In FastSLAM 2.0, the pose $x_t^{[k]}$ is drawn from the posterior

$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t}) \tag{13.26}$$

This distribution differs from the proposal distribution provided in (13.12) in that (13.26) takes the measurement z_t into consideration, along with the correspondence c_t. Specifically, the expression in (13.26) conditions on $z_{1:t}$, whereas the pose sampler in FastSLAM 1.0 conditions on $z_{1:t-1}$.

Unfortunately, it also comes with more complex math. In particular, the mechanism for sampling from (13.26) requires further analysis. First, we rewrite (13.26) in terms of the "known" distributions, such as the measurement and motion models, and the Gaussian feature estimates in the k-th particle.

$$\begin{aligned} (13.27)\quad & p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t}) \\ & \stackrel{\text{Bayes}}{=} \frac{p(z_t \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})\; p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})}{p(z_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})} \\ & = \eta^{[k]}\; p(z_t \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})\; p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \end{aligned}$$

$$
\begin{aligned}
&\overset{\text{Markov}}{=} \quad \eta^{[k]}\; p(z_t \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t});\, p(x_t \mid x_{t-1}^{[k]}, u_t) \\
&= \quad \eta^{[k]} \int p(z_t \mid m_{c_t}, x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \\
&\qquad p(m_{c_t} \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})\; dm_{c_t}\; p(x_t \mid x_{t-1}^{[k]}, u_t) \\
&\overset{\text{Markov}}{=} \quad \eta^{[k]} \int \underbrace{p(z_t \mid m_{c_t}, x_t, c_t)}_{\sim \mathcal{N}(z_t;\, h(m_{c_t}, x_t),\, Q_t)} \quad \underbrace{p(m_{c_t} \mid x_{1:t-1}^{[k]}, z_{1:t-1}, c_{1:t-1})}_{\sim \mathcal{N}(m_{c_t};\, \mu_{c_t,t-1}^{[k]}, \Sigma_{c_t,t-1}^{[k]})}\; dm_{c_t} \\
&\qquad \underbrace{p(x_t \mid x_{t-1}^{[k]}, u_t)}_{\sim \mathcal{N}(x_t;\, g(x_{t-1}^{[k]}, u_t), R_t)}
\end{aligned}
$$

This expression makes apparent that our sampling distribution is truly the convolution of two Gaussians multiplied by a third. In the general SLAM case, the sampling distribution possesses no closed form from which we could easily sample. The culprit is the function h: If it were linear, this probability would be Gaussian, a fact that shall become more obvious below. In fact, not even the integral in (13.27) possesses a closed form solution. For this reason, sampling from the probability (13.27) is difficult.

This observation motivates the replacement of h by a linear approximation. As common in this book, this approximation is obtained through a first order Taylor expansion, given by the following linear function:

(13.28) $$h(m_{c_t}, x_t) \quad \approx \quad \hat{z}_t^{[k]} + H_m (m_{c_t} - \mu_{c_t,t-1}^{[k]}) + H_x\, (x_t - \hat{x}_t^{[k]})$$

Here we use the following abbreviations:

(13.29) $$\hat{z}_t^{[k]} \quad = \quad h(\mu_{c_t,t-1}^{[k]}, \hat{x}_t^{[k]})$$

(13.30) $$\hat{x}_t^{[k]} \quad = \quad g(x_{t-1}^{[k]}, u_t)$$

The matrices H_m and H_x are the Jacobians of h. They are the derivatives of h with respect to m_{c_t} and x_t, respectively, evaluated at the expected values of their arguments:

(13.31) $$H_m \quad = \quad \nabla_{m_{c_t}} h(m_{c_t}, x_t)\Big|_{x_t = \hat{x}_t^{[k]};\, m_{c_t} = \mu_{c_t,t-1}^{[k]}}$$

(13.32) $$H_x \quad = \quad \nabla_{x_t} h(m_{c_t}, x_t)\Big|_{x_t = \hat{x}_t^{[k]};\, m_{c_t} = \mu_{c_t,t-1}^{[k]}}$$

Under this approximation, the desired sampling distribution (13.27) is a Gaussian with the following parameters:

(13.33) $$\Sigma_{x_t}^{[k]} \quad = \quad \left[H_x^T Q_t^{[k]-1} H_x + R_t^{-1} \right]^{-1}$$

(13.34) $$\mu_{x_t}^{[k]} \quad = \quad \Sigma_{x_t}^{[k]} H_x^T Q_t^{[k]-1} (z_t - \hat{z}_t^{[k]}) + \hat{x}_t^{[k]}$$

where the matrix $Q_t^{[k]}$ is defined as follows:

(13.35) $$Q_t^{[k]} = Q_t + H_m \Sigma_{c_t,t-1}^{[k]} H_m{}^T$$

To see, we note that under out linear approximation the convolution theorem provides us with a closed form for the integral term in (13.27):

(13.36) $$\mathcal{N}(z_t; \hat{z}_t^{[k]} + H_x x_t - H_x \hat{x}_t^{[k]}, Q_t^{[k]})$$

The sampling distribution (13.27) is now given by the product of this normal distribution and the rightmost term in (13.27), the normal $\mathcal{N}(x_t; \hat{x}_t^{[k]}, R_t)$. Written in Gaussian form, we have

(13.37) $$p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t}) = \eta \exp\left\{ - P_t^{[k]} \right\}$$

with

(13.38) $$P_t^{[k]} = \frac{1}{2} \Big[(z_t - \hat{z}_t^{[k]} - H_x x_t + H_x \hat{x}_t^{[k]})^T Q_t^{[k]-1} (z_t - \hat{z}_t^{[k]} - H_x x_t + H_x \hat{x}_t^{[k]}) + (x_t - \hat{x}_t^{[k]})^T R_t^{-1} (x_t - \hat{x}_t^{[k]}) \Big]$$

This expression is obviously quadratic in our target variable x_t, hence $p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t})$ is Gaussian. The mean and covariance of this Gaussian are equivalent to the minimum of $P_t^{[k]}$ and its curvature. Those are identified by calculating the first and second derivatives of $P_t^{[k]}$ with respect to x_t:

(13.39) $$\frac{\partial P_t^{[k]}}{\partial x_t} = -H_x^T Q_t^{[k]-1}(z_t - \hat{z}_t^{[k]} - H_x x_t + H_x \hat{x}_t^{[k]}) + R_t^{-1}(x_t - \hat{x}_t^{[k]})$$
$$= (H_x^T Q_t^{[k]-1} H_x + R_t^{-1}) x_t - H_x^T Q_t^{[k]-1}(z_t - \hat{z}_t^{[k]} + H_x \hat{x}_t^{[k]}) - R_t^{-1} \hat{x}_t^{[k]}$$

(13.40) $$\frac{\partial^2 P_t^{[k]}}{\partial x_t^2} = H_x^T Q_t^{[k]-1} H_x + R_t^{-1}$$

The covariance $\Sigma_{x_t}^{[k]}$ of the sampling distribution is now obtained by the inverse of the second derivative

(13.41) $$\Sigma_{x_t}^{[k]} = \left[H_x^T Q_t^{[k]-1} H_x + R_t^{-1} \right]^{-1}$$

The mean $\mu_{x_t}^{[k]}$ of the sample distribution is obtained by setting the first derivative (13.39) to zero. This gives us:

(13.42) $$\mu_{x_t}^{[k]} = \Sigma_{x_t}^{[k]} \left[H_x^T Q_t^{[k]-1}(z_t - \hat{z}_t^{[k]} + H_x \hat{x}_t^{[k]}) + R_t^{-1} \hat{x}_t^{[k]} \right]$$
$$= \Sigma_{x_t}^{[k]} H_x^T Q_t^{[k]-1}(z_t - \hat{z}_t^{[k]}) + \Sigma_{x_t}^{[k]} \left[H_x^T Q_t^{[k]-1} H_x + R_t^{-1} \right] \hat{x}_t^{[k]}$$
$$= \Sigma_{x_t}^{[k]} H_x^T Q_t^{[k]-1}(z_t - \hat{z}_t^{[k]}) + \hat{x}_t^{[k]}$$

This Gaussian is the approximation of the desired sampling distribution (13.26) in FastSLAM 2.0. Obviously, this proposal distribution is quite a bit more involved than the much simpler one for FastSLAM 1.0 in Equation (13.12).

13.4.2 Updating the Observed Feature Estimate

Just like our first version of the FastSLAM algorithm, FastSLAM 2.0 updates the posterior over the feature estimates based on the measurement z_t and the sampled pose $x_t^{[k]}$. The estimates at time $t-1$ are once again represented by the mean $\mu_{j,t-1}^{[k]}$ and the covariance $\Sigma_{j,t-1}^{[k]}$. The updated estimates are $\mu_{j,t}^{[k]}$ and $\Sigma_{j,t}^{[k]}$. The nature of the update depends on whether or not a feature j was observed at time t. For $j \neq c_t$, we already established in Equation (13.4) that the posterior over the feature remains unchanged. This implies that instead of updating the estimated, we merely have to copy it.

For the observed feature $j = c_t$, the situation is more intricate. Equation (13.5) already stated the posterior over observed features. Here we repeat it with the particle index k:

$$\begin{aligned}(13.43)\quad & p(m_{c_t} \mid x_t^{[k]}, c_{1:t}, z_{1:t}) \\ & = \eta \underbrace{p(z_t \mid m_{c_t}, x_t^{[k]}, c_t)}_{\sim \mathcal{N}(z_t;\, h(m_{c_t}, x_t^{[k]}),\, Q_t)} \underbrace{p(m_{c_t} \mid x_{1:t-1}^{[k]}, z_{1:t-1}, c_{1:t-1})}_{\sim \mathcal{N}(m_{c_t};\, \mu_{c_t,t-1}^{[k]}, \Sigma_{c_t,t-1}^{[k]})}\end{aligned}$$

As in (13.27), the nonlinearity of h causes this posterior to be non-Gaussian, which is at odds with FastSLAM 2.0's Gaussian representation for feature estimates. Luckily, the exact same linearization as above provides the solution:

$$(13.44)\quad h(m_{c_t}, x_t) \approx \hat{z}_t^{[k]} + H_m (m_{c_t} - \mu_{c_t,t-1}^{[k]})$$

Notice that x_t is not a free variable here, hence we can omit the third term in (13.28). This approximation renders the probability (13.43) Gaussian in the target variable m_{c_t}:

$$\begin{aligned}(13.45)\quad & p(m_{c_t} \mid x_t^{[k]}, c_{1:t}, z_{1:t}) \\ & = \eta \exp\Big\{ -\tfrac{1}{2}(z_t - \hat{z}_t^{[k]} - H_m (m_{c_t} - \mu_{c_t,t-1}^{[k]}))\; Q_t^{-1} \\ & \qquad (z_t - \hat{z}_t^{[k]} - H_m (m_{c_t} - \mu_{c_t,t-1}^{[k]})) \\ & \qquad -\tfrac{1}{2}(m_{c_t} - \mu_{c_t,t-1}^{[k]})\; \Sigma_{c_t,t-1}^{[k]-1} (m_{c_t} - \mu_{c_t,t-1}^{[k]}) \Big\}\end{aligned}$$

The new mean and covariance are obtained using the standard EKF measurement update equations:

$$K_t^{[k]} = \Sigma_{c_t,t-1}^{[k]} H_m^T Q_t^{[k]-1} \tag{13.46}$$

$$\mu_{c_t,t}^{[k]} = \mu_{c_t,t-1}^{[k]} + K_t^{[k]}(z_t - \hat{z}_t^{[k]}) \tag{13.47}$$

$$\Sigma_{c_t,t}^{[k]} = (I - K_t^{[k]} H_m)\Sigma_{c_t,t-1}^{[k]} \tag{13.48}$$

We notice this is quite a bit more complicated than the update in FastSLAM 1.0, but the additional effort in implementing this often pays out, in terms of improved accuracy.

13.4.3 Calculating Importance Factors

The particles generated thus far do not yet match the desired posterior. In FastSLAM 2.0, the culprit is the normalizer $\eta^{[k]}$ in (13.27), which is usually different for each particle k. These differences are not yet accounted for in the resampling process. As in FastSLAM 1.0, the importance factor is given by the following quotient.

$$w_t^{[k]} = \frac{\text{target distribution}}{\text{proposal distribution}} \tag{13.49}$$

Once again, the target distribution that we would like our particles to assume is given by the path posterior, $p(x_t^{[k]} \mid z_{1:t}, u_{1:t}, c_{1:t})$. Under the asymptotically correct assumptions that paths in $x_{1:t-1}^{[k]}$ have been generated according to the target distribution one time step earlier, $p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1}, c_{1:t-1})$, we note that the proposal distribution is now given by the product

$$p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1}, c_{1:t-1})\; p(x_t^{[k]} \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t}) \tag{13.50}$$

The second term in this product is the pose sampling distribution (13.27). The importance weight is obtained as follows:

$$\begin{aligned} w_t^{[k]} &= \frac{p(x_t^{[k]} \mid u_{1:t}, z_{1:t}, c_{1:t})}{p(x_t^{[k]} \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t})\; p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})} \\ &= \frac{p(x_t^{[k]} \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t})\; p(x_{1:t-1}^{[k]} \mid u_{1:t}, z_{1:t}, c_{1:t})}{p(x_t^{[k]} \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}, c_{1:t})\; p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})} \\ &= \frac{p(x_{1:t-1}^{[k]} \mid u_{1:t}, z_{1:t}, c_{1:t})}{p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})} \end{aligned} \tag{13.51}$$

$$
\begin{aligned}
&\overset{\text{Bayes}}{=} \quad \eta \, \frac{p(z_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \; p(x_{1:t-1}^{[k]} \mid u_{1:t}, z_{1:t-1}, c_{1:t})}{p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})} \\
&\overset{\text{Markov}}{=} \quad \eta \, \frac{p(z_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \; p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})}{p(x_{1:t-1}^{[k]} \mid u_{1:t-1}, z_{1:t-1}, c_{1:t-1})} \\
&= \quad \eta \; p(z_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t})
\end{aligned}
$$

The reader may notice that this expression is the inverse of our normalization constant $\eta^{[k]}$ in (13.27). Further transformations give us the following form:

(13.52)
$$
\begin{aligned}
w_t^{[k]} &= \eta \int p(z_t \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \\
&\qquad p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \; dx_t \\
&\overset{\text{Markov}}{=} \eta \int p(z_t \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \; p(x_t \mid x_{t-1}^{[k]}, u_t) \; dx_t \\
&= \eta \int \int p(z_t \mid m_{c_t}, x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \\
&\qquad p(m_{c_t} \mid x_t, x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t-1}, c_{1:t}) \; dm_{c_t} \; p(x_t \mid x_{t-1}^{[k]}, u_t) \; dx_t \\
&\overset{\text{Markov}}{=} \eta \int \underbrace{p(x_t \mid x_{t-1}^{[k]}, u_t)}_{\sim \mathcal{N}(x_t; g(\hat{x}_{t-1}^{[k]}, u_t), R_t)} \int \underbrace{p(z_t \mid m_{c_t}, x_t, c_t)}_{\sim \mathcal{N}(z_t; h(m_{c_t}, x_t), Q_t)} \\
&\qquad \underbrace{p(m_{c_t} \mid x_{1:t-1}^{[k]}, u_{1:t-1}, z_{1:t-1}, c_{1:t-1})}_{\sim \mathcal{N}(m_{c_t}; \mu_{c_t,t-1}^{[k]}, \Sigma_{c_t,t-1}^{[k]})} \; dm_{c_t} \; dx_t
\end{aligned}
$$

We find that this expression can once again be approximated by a Gaussian over measurements z_t by linearizing h. As it is easily shown, the mean of the resulting Gaussian is $\hat{z}_t$, and its covariance is

(13.53)
$$
L_t^{[t]} = H_x^T \; Q_t \; H_x + H_m \Sigma_{c_t,t-1}^{[k]} \; H_m^T + R_t
$$

Put differently, the (non-normalized) importance factor of the k-th particle is given by the following expression:

(13.54)
$$
w_t^{[k]} = |2\pi L_t^{[t]}|^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}(z_t - \hat{z}_t) \; L_t^{[t]-1} \; (z_t - \hat{z}_t)\right\}
$$

As in FastSLAM 1.0, particles generated in Steps 1 and 2, along with their importance factor calculated in Step 3, are collected in a temporary particle set.

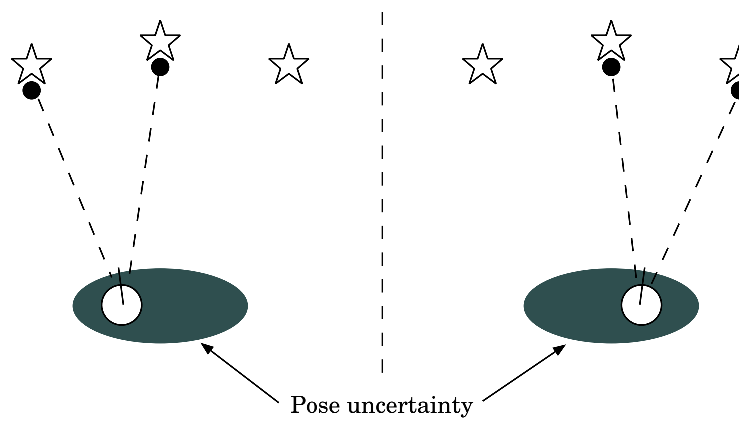

Figure 13.7 The data association problem in SLAM. This figure illustrates that the best data association may vary even within regions of high likelihood for the pose of the robot.

The final step of the FastSLAM 2.0 update is a resampling step. Just as in FastSLAM 1.0, FastSLAM 2.0 draws (with replacement) M particles from the temporary particle set. Each particle is drawn with a probability proportional to its importance factor $w_t^{[k]}$. The resulting particle set represent (asymptotically) the desired factored posterior at time t.

13.5 Unknown Data Association

This section extends both variants of the FastSLAM algorithm to cases where the correspondence variables $c_{1:t}$ are unknown. A key advantage of using particle filters for SLAM is that each particle can rely on its own, local data association decisions.

We remind the reader that the data association problem at time t is the problem of determining the variable c_t based on the available data. This problem is illustrated in Figure 13.7: Here a robot observes two features in the world. Depending on its actual pose relative to these features, these measurements correspond to different features in the map (depicted as stars in Figure 13.7).

So far, we discussed a number of data association technique using arguments such as maximum likelihood. Those techniques had in common that there is only a single data association per measurement, for the entire filter.

FastSLAM, by virtue of using multiple particles, can determine the correspondence on a per-particle basis. Thus, the filter not only samples over robot paths, but also over possible data association decisions along the way.

This is one of the key features of FastSLAM, which sets it aside from the rich body of Gaussian SLAM algorithms. As long as a small subset of the particles is based on the correct data association, data association errors are not as fatal as in EKF approaches. Particles subject to such errors tend to possess inconsistent maps, which increases the probability that they are simply sampled away in future resampling steps.

The mathematical definition of the per-particle data association is straightforward, in that it generalizes the per-filter data association to individual particles. Each particle maintains a local set of data association variables, denoted $\hat{c}_t^{[k]}$. In maximum likelihood data association, each $\hat{c}_t^{[k]}$ is determined by maximizing the likelihood of the measurement z_t:

$$\hat{c}_t^{[k]} = \underset{c_t}{\operatorname{argmax}}\ p(z_t \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}) \tag{13.55}$$

An alternative is the data association sampler (DAS), which samples the data association variable according to their likelihood

$$\hat{c}_t \sim \eta\, p(z_t \mid c_t, \hat{c}_{1:t-1}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}) \tag{13.56}$$

Both techniques, ML and DAS, make it possible to estimate the number of features in the map. SLAM techniques using ML create new features in the map if the likelihood falls below a threshold p_0 for all known features in the map. DAS associates an observed measurement with a new, previously unobserved feature stochastically. They do so with probability proportional to ηp_0, where η is a normalizer defined in (13.56).

$$\hat{c}_t^{[k]} \sim \eta\, p(z_t \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}) \tag{13.57}$$

For both techniques, the likelihood is calculated as follows:

$$\begin{aligned} & p(z_t \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}) \\ &= \int p(z_t \mid m_{c_t}, c_t, \hat{c}_{1:t-1}^{[k]}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t}) \\ &\qquad\qquad p(m_{c_t} \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_{1:t}^{[k]}, z_{1:t-1}, u_{1:t})\, dm_{c_t} \\ &= \int \underbrace{p(z_t \mid m_{c_t}, c_t, x_t^{[k]})}_{\sim \mathcal{N}(z_t; h(m_{c_t}, x_t^{[k]}), Q_t)} \underbrace{p(m_{c_t} \mid \hat{c}_{1:t-1}^{[k]}, x_{1:t-1}^{[k]}, z_{1:t-1})}_{\sim \mathcal{N}(\mu_{c_t,t-1}^{[k]}, \Sigma_{c_t,t-1}^{[k]})}\, dm_{c_t} \end{aligned} \tag{13.58}$$

Linearization of h enables us to obtain this in closed form:

$$p(z_t \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_t^{[k]}, z_{1:t-1}, u_{1:t}) \tag{13.59}$$

$$= \quad |2\pi Q_t^{[k]}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - h(\mu_{c_t,t-1}^{[k]}, x_t^{[k]}))^T Q_t^{[k]-1}(z_t - h(\mu_{c_t,t-1}^{[k]}, x_t^{[k]}))\right\}$$

The variable $Q_t^{[k]}$ was defined in Equation (13.35), as a function of the data association variable c_t. New features are added to the map in exactly the same way as outlined above. In the ML approach, a new feature is added when the probability $p(z_t \mid c_t, \hat{c}_{1:t-1}^{[k]}, x_t^{[k]}, z_{1:t-1}, u_{1:t})$ falls beyond a threshold p_0. The DAS includes the hypothesis that an observation corresponds to a previously unobserved feature in its set of hypotheses, and samples it with probability ηp_0.

13.6 Map Management

Map management in FastSLAM is largely equivalent to EKF SLAM, with a few particulars arising from the fact that data association is handled on a per-particle level in FastSLAM.

As in the alternative SLAM algorithms, any newly added feature requires the initialization of a new Kalman filter. In many SLAM problems the measurement function h is *invertible*. This is the case, for example, for robots measuring range and bearing to features in the plane, in which a single measurement suffices to produce a (non-degenerate) estimate on the feature location. The initialization of the EKF is then straightforward:

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t) \tag{13.60}$$

$$\mu_{n,t}^{[k]} = h^{-1}(z_t, x_t^{[k]}) \tag{13.61}$$

$$\Sigma_{n,t}^{[k]} = (H_{\hat{c}}^{[k]T} Q_t^{-1} H_{\hat{c}}^{[k]})^{-1} \quad \text{with} \quad H_{\hat{c}}^{[k]} = h'(\mu_{n,t}^{[k]}, x_t^{[k]}) \tag{13.62}$$

$$w_t^{[k]} = p_0 \tag{13.63}$$

Notice that for newly observed features, the pose $x_t^{[k]}$ is sampled according to the motion model $p(x_t \mid x_{t-1}^{[k]}, u_t)$. This distribution is equivalent to the FastSLAM sampling distribution (13.26) in situations where no previous location estimate for the observed feature is available.

Initialization techniques for situations in which h is not invertible are discussed in Deans and Hebert (2002). In general, such situations require the accumulation of multiple measurements, to obtain a good estimate for the linearization of h.

To accommodate features introduced erroneously into the map, FastSLAM features a mechanism for eliminating features that are not supported by sufficient evidence. Just as in EKF SLAM, FastSLAM does so by keeping track of the log odds of the actual existence of individual features in the map.

Specifically, when a feature is observed, its log odds for existence is incremented by a fixed amount, which is calculated using the standard Bayes filter formula. Similarly, when a feature is not observed even though it should have, such negative information results in a decrement of the feature existence variable by a fixed amount. Features whose variable sinks below a threshold value are then simply removed from the list of particles. It is also possible to implement a provisional feature list in FastSLAM. Technically this is trivial, since each feature possesses its own particle.

13.7 The FastSLAM Algorithms

Tables 13.2 and 13.3 summarize both FastSLAM variants with unknown data association. In both algorithms, particles are of the form

$$Y_t^{[k]} \quad = \quad \left\langle x_t^{[k]}, N_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, \tau_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_t^{[k]},t}^{[k]}, \Sigma_{N_t^{[k]},t}^{[k]}, \tau_{N_t^{[k]}}^{[k]} \right\rangle \right\rangle \tag{13.64}$$

In addition to the pose $x_t^{[k]}$ and the feature estimates $\mu_{n,t}^{[k]}$ and $\Sigma_{n,t}^{[k]}$, each particle maintains the number of features $N_t^{[k]}$ in its local map, and each feature carries a probabilistic estimate of its existence $\tau_n^{[k]}$. Iterating the filter requires time linear in the maximum number of features $\max_k N_t^{[k]}$ in each map, and it is also linear in the number of particles M. Further below, we will discuss advanced data structure that yield more efficient implementations.

We note that both versions of FastSLAM, as described here, consider a single measurement at a time. As before, this choice is made for notational convenience, and many of the techniques discussed in previous SLAM chapters can be brought to bear.

13.8 Efficient Implementation

At first glance, it may appear that each update in FastSLAM requires $O(MN)$ time, where M is the number of particles M and N the number of features in the map. The linear complexity in M is unavoidable, given that we have to process M particles with each update. The linear complexity in N is the result of the resampling process. Whenever a particle is drawn more than once

Algorithm FastSLAM 1.0(z_t, u_t, Y_{t-1}):

  for $k = 1$ to M do // loop over all particles
    retrieve $\left\langle x_{t-1}^{[k]}, N_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]}, i_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_{t-1}^{[k]},t-1}^{[k]}, \Sigma_{N_{t-1}^{[k]},t-1}^{[k]}, i_{N_{t-1}^{[k]},t-1}^{[k]} \right\rangle \right\rangle$ from Y_{t-1}
    $x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // sample new pose
    for $j = 1$ to $N_{t-1}^{[k]}$ do // measurement likelihoods
      $\hat{z}_j = h(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // measurement prediction
      $H_j = h'(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // calculate Jacobian
      $Q_j = H_j \, \Sigma_{j,t-1}^{[k]} \, H_j^T + Q_t$ // measurement covariance
      $w_j = |2\pi Q_j|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}_j)^T Q_j^{-1}(z_t - \hat{z}_j)\right\}$ // likelihood of correspondence
    endfor
    $w_{1+N_{t-1}^{[k]}} = p_0$ // importance factor, new feature
    $w^{[k]} = \max w_j$ // max likelihood correspondence
    $\hat{c} = \operatorname{argmax} w_j$ // index of ML feature
    $N_t^{[k]} = \max\{N_{t-1}^{[k]}, \hat{c}\}$ // new number of features in map
    for $j = 1$ to $N_t^{[k]}$ do // update Kalman filters
      if $j = \hat{c} = 1 + N_{t-1}^{[k]}$ then // is new feature?
        $\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ // initialize mean
        $H_j = h'(\mu_{j,t}^{[k]}, x_t^{[k]})$; $\Sigma_{j,t}^{[k]} = (H_j^{-1})^T Q_t H_j^{-1}$ // initialize covar.
        $i_{j,t}^{[k]} = 1$ // initialize counter
      else if $j = \hat{c} \leq N_{t-1}^{[k]}$ then // is observed feature?
        $K = \Sigma_{j,t-1}^{[k]} \, H_j^T \, Q_{\hat{c}}^{-1}$ // calculate Kalman gain
        $\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}_{\hat{c}})$ // update mean
        $\Sigma_{j,t}^{[k]} = (I - K \, H_j)\Sigma_{j,t-1}^{[k]}$ // update covariance
        $i_{j,t}^{[k]} = i_{j,t-1}^{[k]} + 1$ // increment counter

see next page for continuation

continued from the previous page

else // *all other features*
$\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]}$ // *copy old mean*
$\Sigma_{j,t}^{[k]} = \Sigma_{j,t-1}^{[k]}$ // *copy old covariance*
if $\mu_{j,t-1}^{[k]}$ *outside perceptual range of* $x_t^{[k]}$ *then* // *should feature have been seen?*
$i_{j,t}^{[k]} = i_{j,t-1}^{[k]}$ // *no, do not change*
else
$i_{j,t}^{[k]} = i_{j,t-1}^{[k]} - 1$ // *yes, decrement counter*
if $i_{j,t-1}^{[k]} < 0$ *then*
discard feature j // *discard dubious features*
endif
endif
endif
endfor
add $\left\langle x_t^{[k]}, N_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, i_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_t^{[k]},t}^{[k]}, \Sigma_{N_t^{[k]},t}^{[k]}, i_{N_t^{[k]}}^{[k]} \right\rangle \right\rangle$ *to* Y_{aux}
endfor
$Y_t = \emptyset$ // *construct new particle set*
do M *times* // *resample* M *particles*
draw random index k *with probability* $\propto w^{[k]}$ // *resample*
add $\left\langle x_t^{[k]}, N_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, i_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_t^{[k]},t}^{[k]}, \Sigma_{N_t^{[k]},t}^{[k]}, i_{N_t^{[k]}}^{[k]} \right\rangle \right\rangle$ *to* Y_t
enddo
return Y_t

Table 13.2 The algorithm FastSLAM 1.0 with unknown data association. This version does not implement any of the efficient tree representations discussed in the chapter.

Algorithm FastSLAM 2.0(z_t, u_t, Y_{t-1}):

for $k = 1$ to M do // loop over all particles

retrieve $\left\langle x_{t-1}^{[k]}, N_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]}, i_1^{[k]} \right\rangle, \ldots,\right.$ $\left. \left\langle \mu_{N_{t-1}^{[k]},t-1}^{[k]}, \Sigma_{N_{t-1}^{[k]},t-1}^{[k]}, i_{N_{t-1}^{[k]}}^{[k]} \right\rangle \right\rangle$ from Y_{t-1}

for $j = 1$ to $N_{t-1}^{[k]}$ do // calculate sampling distribution

$\hat{x}_{j,t} = g(x_{t-1}^{[k]}, u_t)$ // predict pose

$\bar{z}_j = h(\mu_{j,t-1}^{[k]}, \hat{x}_{j,t})$ // predict measurement

$H_{x,j} = \nabla_{x_t} h(\mu_{j,t-1}^{[k]}, \hat{x}_{j,t})$ // Jacobian wrt pose

$H_{m,j} = \nabla_{m_j} h(\mu_{j,t-1}^{[k]}, \hat{x}_{j,t})$ // Jacobian wrt map feature

$Q_j = Q_t + H_{m,j} \Sigma_{j,t-1}^{[k]} H_{m,j}^T$ // measurement information

$\Sigma_{x,j} = \left[H_{x,j}^T Q_j^{-1} H_{x,j} + R_t^{-1} \right]^{-1}$ // Cov of proposal distribution

$\mu_{x_t,j} = \Sigma_{x,j} H_{x,j}^T Q_j^{-1}$ $(z_t - \bar{z}_j) + \hat{x}_{j,t}$ // mean of proposal distribution

$x_{t,j}^{[k]} \sim \mathcal{N}(\mu_{x_t,j}, \Sigma_{x,j})$ // sample pose

$\hat{z}_j = h(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // measurement prediction

$\pi_j = |2\pi Q_j|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\right.$ $\left.(z_t - \hat{z}_j)^T Q_j^{-1} (z_t - \hat{z}_j)\right\}$ // correspondence likelihood

endfor

$\pi_{1+N_{t-1}^{[k]}} = p_0$ // likelihood of new feature

$\hat{c} = \operatorname{argmax} \pi_j$ // ML correspondence

$N_t^{[k]} = \max\{N_{t-1}^{[k]}, \hat{c}\}$ // new number of features

for $j = 1$ to $N_t^{[k]}$ do // update Kalman filters

if $j = \hat{c} = 1 + N_{t-1}^{[k]}$ then // is new feature?

$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // sample pose

$\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ // initialize mean

$H_{m,j} = \nabla_{m_j} h(\mu_{j,t}^{[k]}, x_t^{[k]})$ // Jacobian wrt map feature

$\Sigma_{j,t}^{[k]} = (H_{m,j}^{-1})^T Q_t H_{m,j}^{-1}$ // initialize covariance

$i_{j,t}^{[k]} = 1$ // initialize counter

$w^{[k]} = p_0$ // importance weight

else if $j = \hat{c} \leq N_{t-1}^{[k]}$ then // is observed feature?

$x_t^{[k]} = x_{t,j}^{[k]}$

$K = \Sigma_{j,t-1}^{[k]} H_{m,j}^T Q_j^{-1}$ // calculate Kalman gain

see next page for continuation

continued from the previous page

$\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}_j)$ // *update mean*
$\Sigma_{j,t}^{[k]} = (I - K\ H_{m,j})\ \Sigma_{j,t-1}^{[k]}$ // *update covariance*
$i_{j,t}^{[k]} = i_{j,t-1}^{[k]} + 1$ // *increment counter*
$L = H_{x,j}\ R_t\ H_{x,j}^T + H_{m,j}\ \Sigma_{j,t-1}^{[k]}\ H_{m,j}^T + Q_t$
$w^{[k]} = |2\pi L|^{-\frac{1}{2}}\ \exp\left\{-\frac{1}{2}\right.$
$\left.(z_t - \hat{z}_j)^T\ L^{-1}\ (z_t - \hat{z}_j)\right\}$ // *importance weight*
else // *all other features*
$\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]}$ // *copy old mean*
$\Sigma_{j,t}^{[k]} = \Sigma_{j,t-1}^{[k]}$ // *copy old covariance*
if $\mu_{j,t-1}^{[k]}$ *outside perceptual*
range of $x_t^{[k]}$ *then* // *should feature have been seen?*
$i_{j,t}^{[k]} = i_{j,t-1}^{[k]}$ // *no, do not change*
else
$i_{j,t}^{[k]} = i_{j,t-1}^{[k]} - 1$ // *yes, decrement counter*
if $i_{j,t-1}^{[k]} < 0$ *then*
discard feature j // *discard dubious features*
endif
endif
endif
endfor
add $\left\langle x_t^{[k]}, N_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, i_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_t^{[k]},t}^{[k]}, \Sigma_{N_t^{[k]},t}^{[k]}, i_{N_t^{[k]}}^{[k]} \right\rangle \right\rangle$ *to* Y_{aux}
endfor
$Y_t = \emptyset$ // *construct new particle set*
do M *times* // *resample* M *particles*
draw random index k
with probability $\propto w^{[k]}$ // *resample*
add $\left\langle x_t^{[k]}, N_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]}, i_1^{[k]} \right\rangle, \ldots, \left\langle \mu_{N_t^{[k]},t}^{[k]}, \Sigma_{N_t^{[k]},t}^{[k]}, i_{N_t^{[k]}}^{[k]} \right\rangle \right\rangle$ *to* Y_t
enddo
return Y_t

Table 13.3 The FastSLAM 2.0 Algorithm, stated here unknown data association.