1: **Algorithm Markov_localization(**$bel(x_{t-1}), u_t, z_t, m$**):**
2: *for all* x_t *do*
3: $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}, m)\ bel(x_{t-1})\ dx_{t-1}$
4: $bel(x_t) = \eta\ p(z_t \mid x_t, m)\ \overline{bel}(x_t)$
5: *endfor*
6: *return* $bel(x_t)$

Table 7.1 Markov localization.

terizations that impact the hardness of the problem, such as the information provided by robot measurements and the information lost through motion. Also, symmetric environments are more difficult than asymmetric ones, due to the higher degree of ambiguity.

7.2 Markov Localization

Probabilistic localization algorithms are variants of the Bayes filter. The straightforward application of Bayes filters to the localization problem is called *Markov localization*. Table 7.1 depicts the basic algorithm. This algorithm is derived from the algorithm **Bayes_filter** (Table 2.1 on page 27). Notice that **Markov_localization** also requires a map m as input. The map plays a role in the measurement model $p(z_t \mid x_t, m)$ (line 4). It often, but not always, is incorporated in the motion model $p(x_t \mid u_t, x_{t-1}, m)$ as well (line 3). Just like the Bayes filter, Markov localization transforms a probabilistic belief at time $t-1$ into a belief at time t. Markov localization addresses the global localization problem, the position tracking problem, and the kidnapped robot problem in static environments.

The initial belief, $bel(x_0)$, reflects the initial knowledge of the robot's pose. It is set differently depending on the type of localization problem.

- **Position tracking.** If the initial pose is known, $bel(x_0)$ is initialized by a point-mass distribution. Let $\bar{x}_0$ denote the (known) initial pose. Then

$$bel(x_0) = \begin{cases} 1 & \text{if } x_0 = \bar{x}_0 \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$