2
Fundamentação Teórica

2.1
Conceitos Básicos em Probabilidade

Em robótica probabilística, quantidades, como as medições de um sensor, atuações dos controles, e os estados de um robô e seu ambiente são todos modelados como variáveis aleatórias. *Variáveis Aleatórias* podem assumir vários valores, de acordo com leis probabilísticas específicas. Inferência Probabilística estuda as leis que governam as variáveis aleatórias que normalmente são derivadas de outras variáveis aleatórias e dos dados observados [1], onde

$$p(x) = p(X = x)$$

denota a probabilidade de que a variável aleatória X assuma o valor de x. E, é claro, as probabilidades são sempre não negativas, ou seja $p(x) \geq 0$.

Uma função não negativa utilizada para representar a distribuição de probabilidade caso a variável aleatória seja contínua é a *Função Densidade de Probabilidade* (FDP). Uma função de densidade comum é a distribuição normal com média μ e variância σ^2. Esta distribuição é dada pela função *Gaussiana*

$$p(x) = \frac{1}{\sqrt{2\pi\mu^2}} e^{-\frac{(x-\mu)^2}{\sigma^2}} \tag{2-1}$$

A distribuição normal (2-1) assume que x é um valor escalar. Frequentemente, x é um vetor. Neste caso, são chamadas distribuições normais multivariáveis, dadas por

$$p(x) = \frac{1}{\sqrt{det(2\pi\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{2-2}$$

onde μ é o vetor média e Σ é a matriz de covariância, simétrica (semidefinida positiva). A *Distribuição Conjunta* de duas variáveis aleatórias X e Y é dada por:

$$p(x,y) = p(X = x \quad e \quad Y = y)$$

Esta expressão descreve a probabilidade do evento onde a variável aleatória X

assume o valor de x e Y assume o valor de y. Se X e Y são *Independentes*, temos:

$$p(x,y) = p(x) \cdot p(y)$$

Muitas vezes, as variáveis aleatórias incluem informações sobre outras variáveis aleatórias. Suponha que já sabemos que o valor Y é y, e gostaria de saber a probabilidade do valor de X ser x condicionando a esse fato. Tal probabilidade é chamada de *Probabilidade Condicional*.

$$p(x|y) = p(X = x|Y = y)$$

Se $p(y) > 0$, a probabilidade condicional é definida como:

$$p(x|y) = \frac{p(x,y)}{p(y)} \tag{2-3}$$

Um fato interessante, que resulta da definição de probabilidade condicional e dos axiomas de medidas de probabilidade, é muitas vezes referido como *Teorema da Probabilidade Total* [1]:

$$p(x) = \sum_y p(x|y) \cdot p(y), \quad \text{Discreto} \tag{2-4}$$

$$p(x) = \int p(x|y) \cdot p(y) dy, \quad \text{Contínuo} \tag{2-5}$$

2.1.1 Regra de Bayes

A regra de bayes desempenha um papel predominante na área de robótica probabilística (e inferência probabilística em geral). Ela relaciona condicionais do tipo $p(x|y)$ para seu "inverso", $p(y|x)$. A regra requer que $p(y) > 0$:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2-6}$$

Se x for uma quantidade que gostaríamos de inferir a partir de y, a probabilidade $p(x)$ será referida como *Distribuição de Probabilidade Anterior*. A distribuição $p(x)$ resume o conhecimento que temos sobre X antes de incorporar informação de y. A probabilidade $p(x|y)$ é chamada de *Distribuição de Probabilidade Posterior* sobre X.

Se estamos interessados em inferir uma quantidade x a partir de dados Y, a regra de bayes nos permite fazê-lo através da probabilidade inversa, que especifica a probabilidade de dados y assumindo que x ocorreu. Em robótica, a probabilidade $p(y|x)$ é frequentemente denominada "modelo generativo",

porque descreve, em algum nível de abstração, como o estado da variável X causa a medição do sensor Y.

Como $p(y)$ na equação (2-6) não depende de x, o fator $p(y)^{-1}$ é escrito como um normalizador da Regra de Bayes e é denotado como η, resultando em

$$p(x|y) = \eta p(y|x)p(x) \tag{2-7}$$

2.1.2 Filtro de Bayes para SLAM

A regra de bayes é o arquétipo de inferência probabilística [11], o princípio essencial de praticamente todos os algoritmo de mapeamento.

Suponha que nós queremos aprender sobre uma quantidade x (o mapa), com base em dados de medição d (e.g. Varreduras ou odometria). Então a regra de bayes diz que, o problema pode ser resolvido pela multiplicação de dois termos: $p(x|d)$ e $p(x)$, ver Equação (2-7). O termo $p(d|x)$ especifica a probabilidade de observar a medição d sobre a hipótese de x. Assim, $p(d|x)$ é um modelo generativo, na medida em que descreve o processo de geração das medições do sensor sobre ambientes diferentes m. O termo $p(x)$ (prévio) especifica a probabilidade de assumir que x é o caso no ambiente antes da chegada de todos os dados.

$$p(x|d) = \eta \quad p(d|x)p(x) \tag{2-8}$$

No processo de SLAM há dois tipos diferentes de dados: sensores de medição z_t e controles u_t. Aqui, subscritos são usados como índice de tempo. Em particular, z_t é a medição do sensor tomada no tempo t, e u_t especifica o comando de movimento do robô no intervalo de tempo $[t-1, t]$.

No campo de mapeamento de robôs móveis, é um estimador recursivo para calcular uma sequência de distribuições de probabilidades posteriores sobre as quantidades que não podem ser observadas diretamente, como um mapa ou posição do robô. Este estado desconhecido é representado por x_t.

$$p(x_t|z^t, u^t) = \eta \quad p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z^{t-1}, u^{t-1}) dx_{t-1} \tag{2-9}$$

onde z^t e u^t se refere a dados que conduzem até um tempo t, dados por

$$z^t = \{z_1, z_2, z_3, ..., z_t\}$$

$$u^t = \{u_1, u_2, u_3, ..., u_t\}$$

Notar que o filtro de Bayes é recursivo, isto é, a probabilidade posterior $p(x_t|z^t, u^t)$ é calculado a partir da mesma probabilidade de um tempo anterior. A probabilidade inicial no tempo $t = 0$ é $p(x_0|z^0, u^0) = p(x_0)$.

Se usarmos m para denotar o mapa e R para representar ao robô, da Equação (2-9) obtemos o filtro de bayes:

$$p(R_t, m_t|z^t, u^t) =$$

$$\eta \quad p(z_t|R_t, m_t) \int\int p(R_t, m_t|u_t, R_{t-1}, m_{t-1}) p(R_{t-1}, m_{t-1}|z^{t-1}, u^{t-1}) d\,R_{t-1}\, d\,m_{t-1} \tag{2-10}$$

Para simplificar a Equação (2-11):

- A maioria dos algoritmos de mapeamento assumem que o mundo é estático, o que implica que o índice de tempo pode ser omitido quando se refere ao mapa m.
- A maioria das abordagens assume que o movimento do robô é independente do mapa [11].
- Utilizando a hipótese de Markov, que postula que o estado atual x_t pode ser estimado usando apenas o estado da etapa anterior x_{t-1} [2].

Com essas simplificações, obtém-se uma forma conveniente do filtro de bayes:

$$p(R_t, m|z^t, u^t) = \eta \quad p(z_t|R_t, m) \int p(R_t|u_t, R_{t-1}) p(R_{t-1}, m|z^{t-1}, u^{t-1}) d\,R_{t-1} \tag{2-11}$$

Na equação (2-11) há duas importantes distribuições de probabilidades: $p(R_t|u_t, R_{t-1})$ e $p(z_t|R_t, m)$. Ambos são modelos generativos do robô e seu ambiente.

Modelo de Movimento A probabilidade $p(R_t|u_t, R_{t-1})$ é referida como modelo de movimento, e especifica o efeito de controle u_t no estado R_{t-1}. Descreve a probabilidade de que o controle u_t, se executado no estado R_{t-1}, possa levar ao estado R_t. Devido ao ruído ou efeitos exógenos não modelados. O resultado de um controle será descrito por uma probabilidade posterior. "Um modelo probabilístico adequado pode modelar com precisão os tipos específicos de incerteza que existem na atuação e percepção do robô"[1]. A nossa exposição se concentra totalmente na cinemática de robôs móveis que operam em ambientes planares, a posição de um robô móvel é resumida por três variáveis

$$R = \left(R_x, R_y, R_\theta\right)^T \tag{2-12}$$

onde $\begin{pmatrix} R_x \\ R_y \end{pmatrix}$, é a localização do robô e R_θ é a orientação do robô. Na Figura 2.1 mostra a posição de robô móvel em um sistema de coordenadas globais. Thrun [1] fornece em detalhes dois modelos específicos de movi-

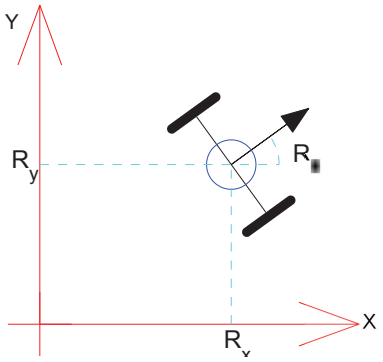


Figura 2.1: Posição e Orientação do Robô

mento probabilísticos, ambos para robôs móveis que operam no plano. O primeiro modelo assume que o controle de movimento u_t especifica um comando de velocidade, determinado pelos motores do robô. O segundo modelo assume que u_t contém informação de odometria (distância percorrida e ângulo de giro).

Modelo de Medição A probabilidade $p(z_t|R_t, m)$ é referida como modelo de medição, porque em termos probabilísticos descreve como as medições do sensor z_t são geradas para diferentes posições R_t e mapas m (um modelo generativo descrevendo o funcionamento dos sensores do robô). O Modelo de Medição conta a incerteza nos sensores do robô. Assim, há explicitamente modelos de ruído na medição do sensor. Na prática, segundo Thrun [1], é muitas vezes impossível modelar um sensor com precisão, principalmente por duas razões: o desenvolvimento de um modelo com precisão para o sensor pode ser extremamente demorado; e um modelo preciso pode exigir variáveis de estado que não são conhecidas, como o material da superfície.

Robótica probabilística acomoda as imprecisões dos modelos de sensores em aspectos estocásticos; modelando o processo de medição como uma densidade de probabilidade condicional, $p(z_t|R_t)$, em vez de uma função determinística $z_t = f(R_t)$, a incerteza no modelo de sensor pode ser

acomodada em aspectos não determinísticos do modelo. Robôs modernos usam uma variedade de tipos de sensores, tais como sensores tácteis, LRF, sensores de sonar ou câmeras. As especificações do modelo dependem do tipo de sensor.

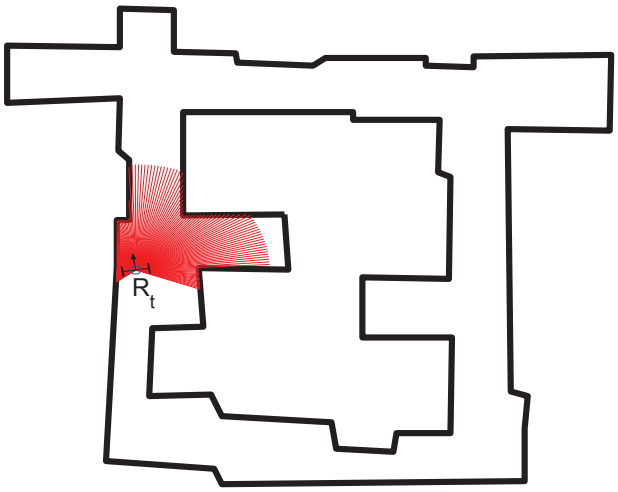


Figura 2.2: Robô móvel em um mapa obtendo as medições a partir do seu LRF.

Observamos também que a Equação de mapeamento (2-11) não pode ser implementada em um computador digital em sua forma global. Isso ocorre porque o posterior sobre o espaço de todos os mapas e posições do robô é uma distribuição de probabilidade sobre um espaço contínuo e, portanto, possui infinitas dimensões. Assim, qualquer algoritmo de mapeamento tem que recorrer a hipóteses adicionais. Estas hipóteses são as principais diferenças entre as diversas soluções para resolver o problema de SLAM.

2.1.3 Laser Range Finder

O desenvolvimento recente de LRFs a preços razoáveis rapidamente o transformou em um sensor dominante para utilização em mapeamento, como na localização de robôs móveis. Com os dados de varredura a laser 2D a aquisição em tempo real, um robô pode-se calcular a área de todo o espaço livre em uma sala, em seguida, ele pode selecionar o centro da sala como a sua posição para a construção do mapa[3], LRF pode dar medições precisas dentro de poucos centímetros, e pode proporcionar leituras muito mais precisas e exatas. Outra diferença importante do LRF é que o laser traça uma linha muito fina através do

ambiente, ver Figura 2.2 mostra o LIDAR $URG04LX-UG01$ [27], gerando leituras de alta resolução.

Figura 2.3: Laser Range Finder (LRF), Hokuyo URG 04LX-UG01.

2.2 Mapa

Um mapa do ambiente é uma lista de objetos no ambiente e suas localizações, e podem ser basicamente classificados em duas categorias: mapas baseados em características (*Feature-Based Maps*) e mapas baseados na Localização (*Ocupancy Grids*).

2.2.1 Representação de mapas por Grade de Ocupação

Oferecem um marcador para qualquer localização no ambiente, e geralmente não assumem restrições geométricas [28]. Visualize um ambiente plano dividido em uma grade regular de quadrados, todos de igual tamanho. Cada um destes quadrados corresponde a uma área física no ambiente, e, como tal, cada quadrado contém um conjunto diferente de objetos ou porções de um objeto. Grade de Ocupação é uma representação abstrata destas seções do mundo, contendo informações sobre se esse quadrado no ambiente real está ocupado ou não. Esta noção de ocupação pode mudar, dependendo da aplicação, mas em geral, quer indicar se a área poderia bloquear a passagem de um robô, ou se a área iria ser detectatda pelos sensores do robô móvel. Idealmente, estes dois conceitos devem os mesmos, mas podem não ser, devido ao ruído e limitações nos sensores. Grade de Ocupação pode ser representada de muitas formas diferentes. É importante assegurar que a representação é bem adaptada ao sensor a ser

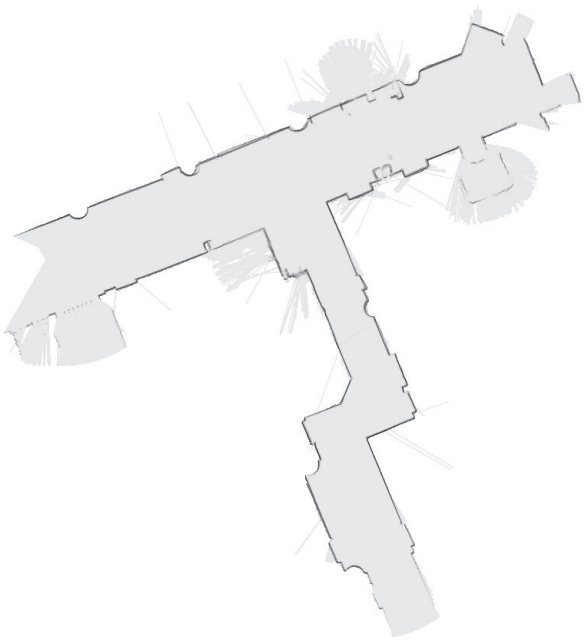


Figura 2.4: Representação de um ambiente mediante Grade de Ocupação

utilizado e à aplicação. A maioria das abordagens tradicionais pode ser generalizada em dois tipos: determinísticos e estocásticos [4].

Mapas Determinísticos São a representação mais simples, e têm um conjunto discreto de valores para as quadrículas. Estes geralmente são **VAZIOS** ou **OCUPADOS**, e às vezes podem incluir um valor para **DESCONHECIDO** ou **NÃO OBSERVADO**.

Mapas Estocásticos Têm uma noção de **OCUPADOS** e **VAZIOS**, assim como mapas deterministicos, mas em vez de ver estes valores como absolutos, eles têm uma escala de vários graus de ocupação. Qual a porcentagem do quadrado é acreditado para ser ocupado, ou o quanto um objeto é transparente para o sensor, são alguns dos fatores que afetam o valor de ocupação.

A representação estocástica e o modelo de observação correspondente precisam ser devidamente calibrados para o sensor utilizado.

Estes mapas têm dois grandes inconvenientes para a realização de localização. Primeiro, é computacionalmente caro combinar tais representações densas do espaço. Em segundo lugar, a própria granularidade de uma representação da Grade pode produzir estimativas de baixa resolução da posição do robô móvel.

2.2.2 Representação de mapas baseado nas Características

São constituídos por um conjunto de características, juntamente com a sua localização cartesiana, e são amplamente utilizados no contexto de localização. No entanto, estes mapas introduzem restrições geométricas, tais como a existência de linhas e cantos no ambiente [29].

A principal vantagem desse tipo de mapas é a sua compacta representação. Por outro lado, requer a existência de estruturas ou objetos que se distinguem uns dos outros o suficiente, e portanto não são adequados para modelar ambientes não estruturados.

Assim, é necessário um algoritmo extra para reconhecer e detectar características [2]. Na prática, os pontos de referência podem ter características semelhantes, o que frequentemente torna difícil distinguir um do outro. Quando isso acontece, o problema da associação de dados, também conhecido como o problema de correspondência, tem de ser tratado. *"O problema de correspondência seria determinar se as medições do sensor tomadas em diferentes pontos no tempo correspondem ao mesmo objeto físico no mundo"* [11]. É um problema difícil, porque o número de hipóteses possíveis pode crescer exponencialmente.

2.3 Interação do Robô com o Ambiente

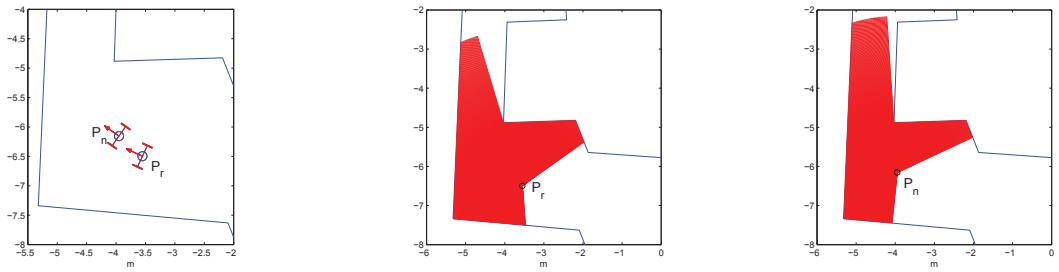
Existem dois tipos fundamentais de interações entre um robô móvel e seu ambiente: o robô pode influenciar a percepção do seu ambiente através de seus atuadores, e pode adquirir informações de seu ambiente através de seus sensores. A percepção é o processo pelo qual o robô utiliza seus sensores para obter informações sobre as características de seu ambiente. Por exemplo, um robô móvel pode levar uma câmera, um LRF , ou consultar seus sensores tácteis para receber informações sobre o ambiente. Os atuadores controlam as ações do robô móvel para alterar sua percepção do ambiente. Exemplos de ações incluem controlar o movimento do robô móvel e a manipulação de objetos, ver Figura 1.1.

2.4 Correspondência de Varreduras (Scan Matching)

Correspondência de Varreduras é o problema de encontrar a translação e rotação $(\triangle x, \triangle y, \triangle \theta)$ entre uma varredura de entrada e uma varredura de referência obtidos a partir de um dispositivo como o LRF, de modo que eles se

sobreponham maximamente [30].

Na Figura 2.5, se apresenta um robô móvel que começa na posição P_r (posição de referência), executa uma varredura S_{ref} (varredura de referência), se move através de um ambiente estático para uma nova posição P_n (posição nova) e executa outra varredura S_{atu} (varredura atual). Os algoritmos de Correspondência de Varreduras procuram então a diferença entre a posição P_n e a posição P_r. A maioria destes algoritmos precisam de um alinhamento inicial entre duas varreduras do sensor, que é fornecida pela leitura de odometria em muitos casos.



(a) (b) (c)

Figura 2.5: Posição Inicial e Final do robô móvel.

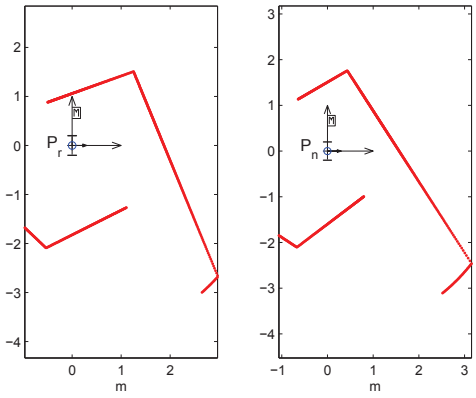


Figura 2.6: Varreduras geradas pelo sensor.

2.4.1 Algoritmo de Correspondência de Varreduras por NDT

As correspondências assumidas entre dois Scans de duas diferentes poses do robô móvel sempre apresentam erros. Biber [17] propôe uma representação alternativa para uma varredura, semelhante a uma grade de ocupação, subdividindo o plano $2D$ em grades. Para cada grade, se atribui uma distribuição normal, que localmente modela a probabilidade de medição de um ponto. *O re-*

sultado da transformação é uma parte contínua de uma densidade de probabilidade diferenciável [17]. Apresenta-se a seguir o algoritmo em detalhes e mostra-se uma aplicação para resolver o problema do SLAM.

2.4.1.1 Notação para o algoritmo

O algoritmo Correspondência de Varreduras requer dois conjuntos de leituras do LRF chamados Varreduras. $S_{ref} = \{q_1, q_2, ..., q_3\}$ representa um conjunto de n pontos que se reuniram no sistema de coordenada A, que é chamado **Varredura de referência**. $S_{atu} = \{p_1, p_2, ..., p_3\}$ representa um conjunto de m pontos que se reuniram no sistema de coordenada B, que é chamado **Varredura atual**.

O objetivo do algoritmo NDT é a de estimar o movimento do robô entre os sistemas de coordenadas A e B. x_B^A denota a estimação da posição relativa entre os sistemas de coordenadas A e B. Assim, x_B^A representa a Rotação e Translação no plano entre A e B. Uma suposição comum é que o erro na correspondência é Normal [17]. Assim, representamos a estimativa do algoritmo NDT como uma distribuição normal multivariável.

$$x_B^A = N(\hat{x}_B^A, P_B^A) \tag{2-13}$$

Aqui, $\hat{x}_B^A$ denota a média do vetor X que tem a forma $[\triangle x, \triangle y, \triangle \theta]$, onde $\triangle x$ e $\triangle y$ representam a translação e $\triangle \theta$ representa a rotação. Por conseguinte, a covariância P_B^A é uma matriz $3x3$.

Para lidar com os processos corrompidos por ruído Gaussiano, uma abordagem comum no mapeamento estocástico e SLAM é o uso do operador $\bigoplus$ (composição de transformações). Este operador é usado ao longo deste trabalho.

$S'_{atu} = \{p'_1, p'_2, ..., p'_3\}$ representa um conjunto de pontos projetados no sistema de coordenadas de S_{ref}. Isto é:

$$p'_i = x_B^A \oplus p_i, \qquad \forall p_i \in S_{atu}$$

O operador composição de transformações $\bigoplus$ representa a transformação espacial entre dois sistemas de coordenadas, que é dada por:

$$x_B^A \oplus p_i = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2-14}$$

onde t_x e t_y descrevem a translação e ϕ a rotação entre as duas posições do robô móvel, (x,y) representa um ponto p_i do S_{atu}, (x',y') representa um ponto do S_{atu} projetado no sistema de coordenadas do S_{ref}, e

$$x_A^B = \begin{pmatrix} t_x \\ t_y \\ \phi \end{pmatrix} \tag{2-15}$$

A ideia do algoritmo NDT é encontrar o movimento do robô x_B^A que maximiza a sobreposição entre as partes do meio ambiente representadas por S_{ref} e S_{atu}.

No contexto do presente trabalho, a função de probabilidade pode ser definida como $f : \Re^2 \rightarrow \Re$, calculando a probabilidade de ter uma leitura de uma Varredura, no sistema de coordenadas $x-y$. Além disso, esta função é construída utilizando as leituras do LRF no S_{ref} , e não em mapas a priori.

É importante ressaltar que a função $f(x,y)$ não é uma FDP (Função Densidade de Probabilidade). No entanto, transformar $f(x,y)$ em uma FDP só envolve o uso de um fator de normalização η. Para entender completamente a utilidade de $f(x,y)$ no algoritmo de Correspondência de Varreduras, é definido a FDP $g(x,y)$ como:

$$g(x,y) = \eta f(x,y) \tag{2-16}$$

Mediante está função pode-se gerar o conjunto de pontos S'_{atu}, onde cada ponto $p'_i \in S'_{atu}$ foi gerado pelo desenho da FDP definida pela Equação (2-16). Sendo S'_{atu} o resultado da composição x_B^A com cada ponto de S_{atu}, a solução de Correspondência de Varreduras pode ser vista como o de encontrar x_B^A que torna a suposição do processo generativo verdade. De um ponto de vista probabilístico, isto pode ser expresso como o problema de maximizar a função de verossimilhança

$$\Psi(x) = \prod_{p_i \in S_{atu}} g(x \oplus p_i) \tag{2-17}$$

onde x representa a rototranslação entre os sistemas de coordenadas de S_{ref} e S_{atu}. A ideia por trás desta função (2-17) é projetar cada ponto da Varredura atual sobre a FDP g por meio da rototranslação x. Em seguida, a FDP é avaliada em cada um destes pontos projetados e os resultados são multiplicados. Uma boa rototranslação x vai projetar os pontos de S_{atu} sobre regiões de $G(x,y)$ com valores elevados (isto é, com elevada probabilidade de ter uma leitura do sensor). Assim, quanto melhor a rototranslação x, maiores os valores da função (2-17).

A rototranslação x que maximiza a Equação (2-17) constitui a estimativa de Correspondência de Varreduras x_B^A. Em consequência, a função representa a sobreposição entre os duas Varreduras.

Como maximizar a Equação (2-17) pode ser computacionalmente caro, uma abordagem usual é a utilização das funções *log-likelihood*. Assim, o problema de Correspondência de Varreduras pode ser expressado como a maximização da função *log-likelihood*

$$F(x) = log(\Psi(x))$$

$$F(x) = \sum_{p_i \in S_{atu}} log(g(x \oplus p_i))$$

$$F(x) = m\,log(\eta) + \sum_{p_i \in S_{atu}} log(f(x \oplus p_i)) \tag{2-18}$$

onde m é o número de pontos em S_{atu}, e η representa um valor constante que não vai influenciar no processo de maximização. Assim, não é necessário considerar η, simplificando para obter a função a ser maximizada

$$F(x) = \sum_{p_i \in S_{atu}} log(f(x \oplus p_i)) \tag{2-19}$$

2.4.1.2 Função de Otimização

O processo de otimização consiste em maximizar a Equação (2-19). Como os problemas de otimização são normalmente descritos como problemas de minimização, adotaremos nossa notação esta convenção. Assim, a função a ser minimizada é $-F(x)$, que chamaremos **Função Objetivo**:

$$score(x) = -\sum_{p_i \in S_{atu}} log(f(x \oplus p_i)) \tag{2-20}$$

2.4.1.3 Transformada de Distribuíções Normais (NDT)

A NDT modela a distribuição de todos os pontos $2D$ reconstruídos de uma Varredura dado pelo LRF, por um conjunto de distribuições normais locais. A NDT é construída a partir de $q_i \in S_{ref}$. Em primeiro lugar, o espaço $2D$ em torno do robô subdivide-se regularmente em N quadrículas com tamanho constante LxL. Em seguida, procura-se o conjunto de pontos de q_i dentro de cada quadrícula. Biber [17] e Burguera [31] propõem um valor de $L = 1m$. Dependendo da posição de origem da quadrícula no intervalo $[0, L)x[0, L)$, os

pontos de S_{ref} serão divididos em grupos diferentes. Vamos denotar por α uma posição particular da origem da quadrícula. Depois, para cada quadrícula $j = 1,2,3,...,N$, que contém pelo menos três pontos, é feito o seguinte:

1. Reúna o conjunto de n pontos $q_i \in S_{ref}$ contida nesta quadrícula em $\Omega_{\alpha,j}$.

2. Calcule a média $\mu_{\alpha,j}$ e a matriz de covariância $P_{\alpha,j}$ dos pontos $q_i \in \Omega_{\alpha,j}$:

$$\mu_{\alpha,j} = \frac{1}{n}\sum_i q_i \tag{2-21}$$

$$P_{\alpha,j} = \frac{1}{n}\sum_i (q_i - \mu_{\alpha,j})(q_i - \mu_{\alpha,j})^T \tag{2-22}$$

   para $i = 1,2,...,n$

3. Para evitar que a matriz de covariância $P_{\alpha,j}$ seja singular ou esteja próximo de ser singular, o menor autovalor de $P_{\alpha,j}$ é testado para ser pelo menos $0,001$ vezes o maior autovalor. Caso contrário, ele é definido para possuir este valor. O parâmetro $0,001$ foi experimentalmente ajustado em [17].

4. Modelar a probabilidade de ter uma leitura no ponto x contida na quadrícula j pela distribuição normal $N(\mu_{\alpha,j}, P_{\alpha,j})$. Deixando o fator de normalização η fora da FDP (2-16), a função de Probabilidade correspondente à quadrícula j é

$$f_{\alpha,j}(x) = e^{-\frac{(x-\mu_{\alpha,j})^T P_{\alpha,j}^{-1}(x-\mu_{\alpha,j})}{2}} \tag{2-23}$$

Assim, a função de probabilidade $f_\alpha(x)$ associada ao S_{ref} e à origem da quadrícula α é construída usando a Equação (2-23), como segue:

$$f_\alpha(x) = \begin{cases} f_{\alpha,1}(x), & \text{se x} \in \Omega_{\alpha,1}\ , \\ f_{\alpha,2}(x), & \text{se x} \in \Omega_{\alpha,2}, \\ ... & \\ f_{\alpha,N}(x), & \text{se x} \in \Omega_{\alpha,N}, \end{cases} \tag{2-24}$$

Para minimizar os efeitos da discretização, a abordagem original do NDT [17] propõe a utilização de quatro quadrículas que se sobrepõem, em vez de usar 1 quadrícula. Agora, cada ponto cai em quatro quadrículas. Assim, quatro funções de probabilidade são construídas, cada uma delas considerando uma posição particular de origem da quadrícula. Dado um ponto x, quatro valores de probabilidade, $f_1(x)$, $f_2(x)$, $f_3(x)$ e $f_4(x)$, estão disponíveis. Usando essa abordagem, para avaliar a função de probabilidade de x, as contribuições das quatro quadrículas são somadas

$$f(x) = \sum_{1 \le \alpha \le 4} f_\alpha(x)$$

Quando $f(x)$ é definido, o processo de otimização tem que ser realizado. De acordo com à Equação (2-20), a função a ser otimizada na abordagem NDT aplicando sobreposição deve ser

$$score(x) = -\sum_{p_i \in S_{atu}} log(\sum_{1 \leq \alpha \leq 4} f_\alpha(x \oplus p_i)) \tag{2-25}$$

No entanto, em vez de minimizar $score(x)$, tal como definido na Equação (2-25), a NDT original [17] minimiza a seguinte função, $h(x)$:

$$h(x) = -\sum_{p_i \in S_{atu}} \sum_{1 \leq \alpha \leq 4} f_\alpha(x \oplus p_i) \tag{2-26}$$

A principal vantagem de definir a função $h(x)$ como a função a ser otimizada, em vez da função $score(x)$ (2-25), é porque $h(x)$ torna o processo de otimização mais fácil e mais rápido. Além disso, Biber [32] mostra que $h(x)$ é uma boa aproximação de $score(x)$.

2.4.1.4 Esboço do Algoritmo Correspondência de Varreduras por NDT

O algoritmo de Correspondência de Varreduras por NDT pode ser esboçado pelas etapas:

1. Construir o NDT do S_{ref} para determinar a função de probabilidade (2-24) associada ao S_{ref}.

2. Estimar um valor inicial para as variáveis (t_x, t_y, ϕ) (nulo ou usando dados de odometria).

3. Para cada ponto $p_i = (x_i, y_i)$ do S_{atu}, determinar o ponto $p'_i = (x'_i, y'_i)$ que representa a transformação espacial $2D$ no sistema de coordenadas do S_{ref} mediante a Equação (2-14), de acordo com os parâmetros (t_x, t_y, ϕ).

4. Avaliar cada ponto p'_i com a função de probabilidade associada ao S_{ref}, e determinar a função de otimização (2-26).

5. Calcular uma estimativa nova para os parâmetros (t_x, t_y, ϕ), tentando otimizar a função de otimização (2-26).

6. Retornar à etapa 3 até que o critério de convergência seja verdade.

2.5 Inteligência Computacional

È um ramo da Ciência da Computação empregada, principalmente, na solução de problemas para os quais não existem procedimentos efeitivos capazes de solucioná-los satisfatoriamente. Problemas assim, geralmente podem ser modelados como tarefas de aprendizagem, percepção, previsão, adaptação ou evolução, e estas são as principais características presentes nas técnicas de Inteligência Computacional. Atualmente a Inteligência Computacional compreende um grande conjunto de técnicas, mas as de maior destaque são as Redes Neurais Artificiais, a Lógica Nebulosa e os Algoritmos Genéticos.

Os Algoritmos Genéticos (AG) pertencem ao grupo dos métodos evolutivos que são baseados em po´pulações de potenciais soluções de um problema. Métodos dessa natureza se mostram interessantes na resolução de problemas complexos de otimização porque conseguem um equilíbrio entre capacidade de exploração do espaço de soluções e também de aproveitamento das melhores soluçõesn ao longo da evolução.

2.6 Evolução Diferencial

Evolução Diferencial (ED) é um algoritmo evolutivo, derivado de um AG com seleção por torneio ,desenvolvido para otimização numérica global. ED tem sido aplicado a uma grande variedade de tarefas de otimização, frequentemente com grande sucesso [33]. Em termos mais simples, otimização é a tentativa de maximizar as propriedades desejáveis de um sistema ao mesmo tempo, e minimizar suas características indesejáveis.

Em geral, para otimizar certas propriedades do sistema através de parâmetros do mesmo, esses parâmetros são geralmente representados por um vetor. O primeiro passo de um problema de otimização começa por elaborar uma função objetivo, que pode modelar os objetivos do problema, incorporando quaisquer restrições. Como quase todos os algoritmos evolutivos, ED é um otimizador baseado na população que, como ponto de partida, ataca o problema por amostragem da função objetivo em N_P vezes, escolhendo aleatoriamente pontos iniciais.

Cada vetor é indexado por um número de 0 a $N_P - 1$. ED gera novos pontos que são perturbações de pontos existentes. Essa perturbação dos vetores é feita por uma diferença de dois vetores de populações selecionadas aleatoriamente.

Para produzir um vetor intermediário u_i, ED adiciona o vetor diferença a um terceiro vetor da população selecionado aleatoriamente. Na etapa de seleção, o vetor intermediário u_i compete com o vetor população do mesmo índice, que neste caso é o número i ($i = 0,1,2,...,N_P - 1$). O vetor com o menor valor da função objetivo é marcado como um membro da próxima geração. Os N_P sobreviventes das competições se tornam pais para a próxima geração do ciclo evolutivo.

2.6.1 Estrutura da População

A implementação mais versátil de ED mantém um par de populações de vetores, ambos os quais contêm N_P vetores D-dimensionais de parâmetros de valor real.

$$P_{x,g} = [\mathbf{x}_{i,g}], \qquad i = 0,1,2,...,N_P - 1, \quad g = 0,1,2,...,g_{max} \tag{2-27}$$

$$\mathbf{x}_{i,g} = [x_{j,i,g}], \qquad j = 0,1,2,...,D-1$$

onde o índice g indica a geração a que vetor pertence, $P_{x,g}$ é a população na geração g, a cada vetor é atribuído um índice de população i, e os parâmetros dentro de cada vetor são indexados com j.

Uma vez inicializado, é realizada a etapa de mutação dos vetores escolhidos aleatoriamente para produzir uma população intermediária:

$$P_{v,g} = [\mathbf{v}_{i,g}] \tag{2-28}$$

$$\mathbf{v}_{i,g} = [v_{j,i,g}]$$

Cada vetor na população atual é recombinado com a população intermediária para produzir uma população

$$P_{u,g} = [\mathbf{u}_{i,g}] \tag{2-29}$$

$$\mathbf{u}_{i,g} = [u_{j,i,g}]$$

2.6.2 Etapas do Algoritmo de ED

2.6.2.1 Inicialização

Antes que a população seja inicializada, os limites superiores e inferiores para cada parâmetro devem ser especificados como o vetor (b_L, b_U), onde b_L e

b_U indicam os limites inferior e superior, respectivamente. Assim, um gerador de números aleatórios atribui a cada parâmetro de cada vetor um valor dentro da faixa prevista. Por exemplo, o valor inicial ($g = 0$) do parâmetro j do vetor população de ordem i é:

$$\mathbf{x}_{j,i,0} = rand_j(0,1) * (b_{j,U} - b_{j,L}) + b_{j,L} \tag{2-30}$$

2.6.2.2 Mutação

Uma vez inicializado, ED faz a mutação e recombina a população para produzir uma população de N_P vetores intermediários

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \tag{2-31}$$

onde F é o fator de escala que controla a taxa em que a população evolui $F \in [0,1]$ [33]. Os índices $r1$ e $r2$ são também escolhidos aleatoriamente, uma vez por mutação, o índice $r0$, pode ser determinada numa variedade de formas (aleatoriamente, maior valor da função objetivo).

2.6.2.3 Crossover

ED também emprega crossover uniforme, às vezes referido como recombinação discreta. Em particular, ED cruza cada vetor com um vetor mutante:

$$\mathbf{u}_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{Se } (rand_j(0,1) \leq Cr \| j = j_{rand}) \\ x_{j,i,g}, & \text{outro caso} \end{cases} \tag{2-32}$$

A probabilidade de crossover, $Cr \in [0,1]$, é um valor definido pelo usuário que controla a fração dos valores atribuídos a cada parâmetro que são copiados a partir do mutante.

2.6.2.4 Seleção

Se o vetor intermediário $u_{i,g}$ tem um valor de função objetivo igual ou menor do que o objetivo de seu vetor do qual herda parâmetros, $x_{i,g}$, ele substitui o vetor na seguinte geração, caso contrário o vetor mantém o seu lugar na população pelo menos por mais uma geração.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, & \text{Se } (f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})) \\ \mathbf{x}_{i,g}, & \text{outro caso} \end{cases} \tag{2-33}$$

Uma vez que a nova população esteja calculada, o processo de recombinação, mutação e seleção é repetido até que um ótimo seja atingido, ou um critério de parada pré-especificado seja satisfeito, por exemplo, o número de gerações atingindo um máximo pré-definido g_{max} [33]. A Figura 2.7 mostra o processo do algoritmo de evolução diferencial.

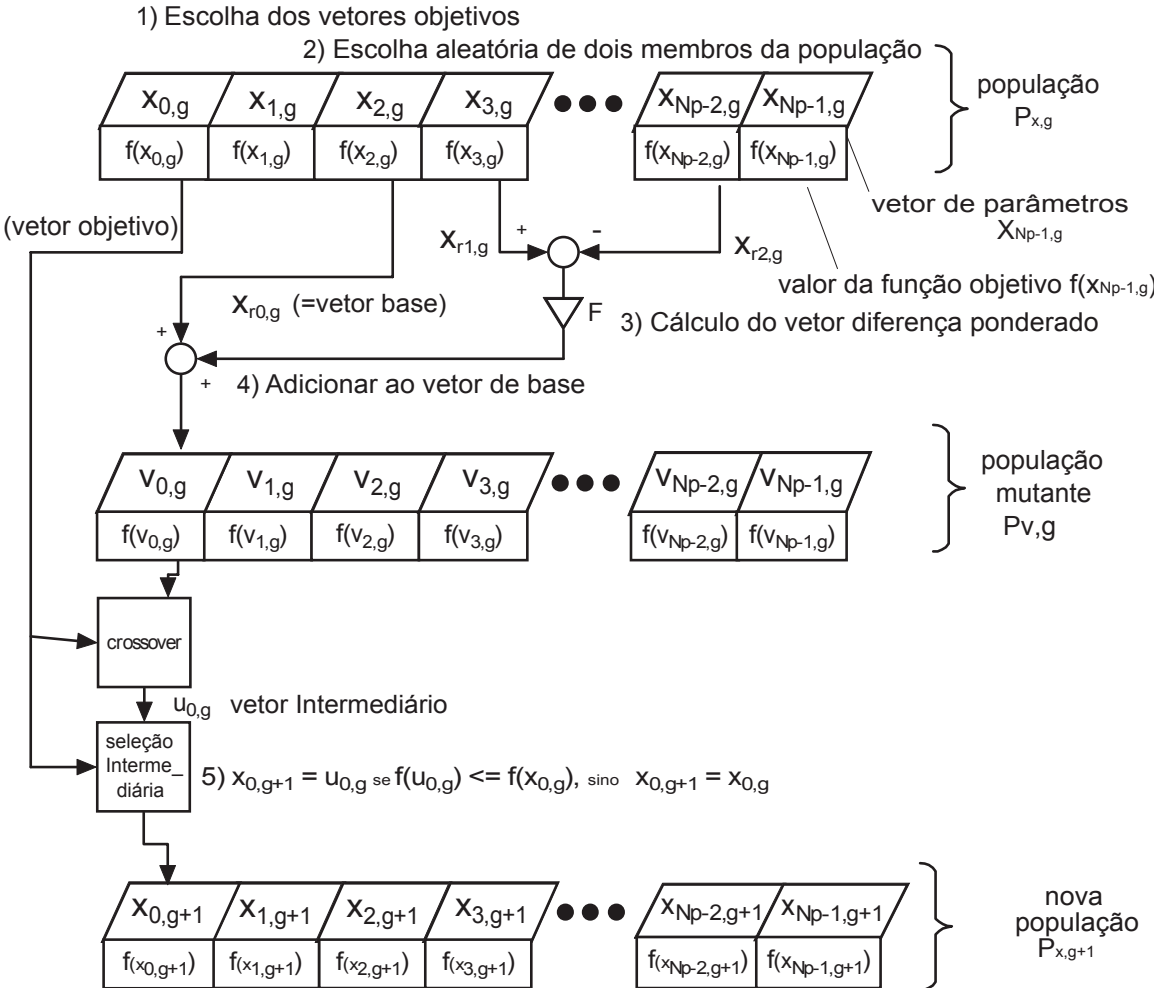


Figura 2.7: Processo de Evolução Diferencial

2.7 DP-SLAM

O algoritmo de DP-SLAM fornece uma solução precisa para resolver o problema de SLAM, mantendo de forma eficaz uma distribuição conjunta através de mapas e as posições do robô móvel usando filtro de partículas, fornece mapas precisos [21]. Quando se utiliza um filtro de partículas para resolver o problema de SLAM, cada partícula corresponde a uma trajetória específica através do ambiente, e possui um mapa específico associado a ele. Quando uma partícula é reamostrada, o mapa inteiro é tratado como parte do estado oculto que está

sendo monitorado.

Para que um filtro de partículas acompanhe corretamente a distribuição conjunta sobre a posição do robô móvel e o mapa, é necessário para cada partícula manter um mapa separado. Durante a fase de reamostragem do filtro de partículas, cada partícula pode ser reamostrada várias vezes. Por conseguinte, é necessário copiar o mapa inteiro para cada nova partícula, para manter cada hipótese do conjunto de atualizações do mapa. Esta distribuição conjunta, e a correspondente necessidade de múltiplas hipóteses de mapas, é ponto de partida crucial para o processo de SLAM [4].

DP-SLAM usa uma representação eficiente para fazer cópias de mapas, ao mesmo tempo reduzindo a memória total exigida para representar este grande número de grades de ocupação. Isto é conseguido através de um método de mapeamento chamado partículas distribuídas (PD-Mapping), o qual explora as redundâncias significativas entre os diferentes mapas. Além disso, é capaz de manter e atualizar centenas de mapas candidatos e posições do robô móvel de uma forma eficiente [22].

2.7.1 Filtro de Partículas

O filtro de partículas (FP) é uma implementação alternativa não paramétrica do filtro de bayes. Ele não depende de uma forma fixa funcional da distribuição de probabilidade, tal como as gaussianas. Filtros de partículas aproximam a distribuição por um número finito de valores.

A ideia principal do FP é representar uma função de distribuição posterior $p(x_t)$ por um conjunto aleatório de amostras retiradas a partir desta distribuição [1]. A Figura 2.8 mostra essa idéia para uma distribuição Gaussiana. Em vez de representar a distribuição de uma forma paramétrica, o FP representa uma distribuição de um conjunto de amostras desta Gaussiana. À medida que o número de amostras vai para o infinito, o FP tende a convergir uniformemente para a distribuição correta. Em FP, as amostras da distribuição $p(x_t)$ são chamadas de *partículas*. Assim, a distribuição de $p(x_t)$ é representada por M partículas ponderadas:

$$\Phi_t := \{\langle x_t^i, w_t^i \rangle / i = 1, 2, ..., M\} \tag{2-34}$$

A intuição por trás de filtros de partículas é aproximar a distribuição $p(x_t|z^t, u^t)$ pelo conjunto de partículas Φ_t. Assim a aproximação feita por FP é dada por:

$$p(x_t|z^t, u^t) \approx \Phi_t \tag{2-35}$$

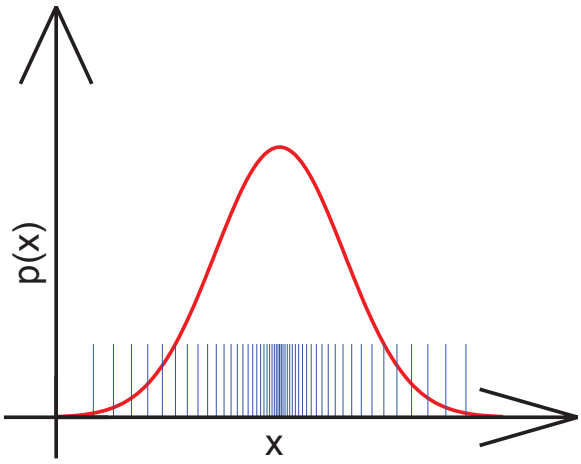


Figura 2.8: Representação da Gaussiana por um conjunto de partículas

O algoritmo de FP constrói a distribuição $p(x_t|z^t, u^t)$ recursivamente a partir da distribuição $p(x_{t-1}|z^{t-1}, u^{t-1})$ de um tempo anterior. Isto significa que os FP podem construir um conjunto de partículas Φ_t recursivamente a partir do conjunto de partículas Φ_{t-1} [1].

A variante mais básica do algoritmo FP é indicada na Tabela 2.1. A entrada deste algoritmo é o conjunto de partículas Φ_{t-1}, junto com o controle mais recente u_t com a mais recente medição z_t.

2.7.2 Representação do Mapa

DP-SLAM utiliza uma representação para o mapa do tipo Grade de Ocupação, mais especificamente por mapas estocásticos, onde cada quadrado da grade tem uma escala de vários graus de ocupação.

DP-SLAM concentra-se em um modelo de LRF. Com as características do LRF um método mais adequado para representar a incerteza do mapa, que leva em conta a distância que o laser viaja através de cada quadrado da grade. O pressuposto básico é que cada quadrado da grade responde à luz do laser de uma forma que é independente dos quadrados vizinhos. Além disso, o comportamento não depende do ângulo de incidência ou a parte do quadrado que é atingido. O comportamento em relação a um quadrado particular depende apenas da distância que o laser percorre até o quadrado.

Tabela 2.1: Algoritmo de Filtro de Partículas [1]

Algoritmo $Particle\_filter(\Phi_{t-1}, u_t, z_t)$	
1:	$\overline{\Phi}_t = \Phi_t = 0$
2:	for $i = 1$ to M do
3:	sorteie $x_t^i \sim p(x_t \mid u_t, x_{t-1}^i)$
4:	$w_t^i = p(z_t \mid x_t^i)$
5:	$\overline{\Phi}_t = \overline{\Phi}_t + \langle x_t^i, w_t^i \rangle$
6:	end for
7:	for $i = 1$ to M do
8:	sorteie i com probabilidade $\propto w_t^i$
9:	some x_t^i a Φ_t
10:	end for
11:	retornar Φ_t

O modelo corresponde à suposição de que, entre as varreduras, as obstruções em cada quadrado da grade são redistribuídas de maneira uniforme dentro do quadrado. Na representação por Grade de Ocupação com ocupação parcial, cada grade é um potencial obstáculo [2].

Seja $P_c(x,\rho)$ a probabilidade cumulativa do laser ter sido interrompido depois de viajar à distância x através de um meio de tipo ρ. Esta condição de consistência pode ser indicada de modo mais geral, em termos de k divisões como [21]:

$$P_c(x,\rho) = \sum_{i=1}^{k} P_c(\frac{x}{k},\rho)(1 - P_c(\frac{x}{k},\rho))^{i-1} \tag{2-36}$$

Esta soma representa que o laser pode ser interrompido durante todo o segmento de comprimento $\frac{x}{k}$, acumulando a probabilidade de parada em cada um. Dentro da soma, o primeiro termo é a probabilidade de que a varredura será obstruído no segmento dado. O segundo termo representa a probabilidade de cada segmento anterior não obstruir ao laser.

A distribuição exponencial $P_c(x,\rho) = 1 - e^{-\frac{x}{\rho}}$, para um escalar positivo ρ, satisfaz essa condição de consistência, onde ρ refere como a opacidade do quadrado da grade.

A probabilidade de um raio inteiro do laser ter sido parado em algum ponto

ao longo de sua trajetória é, portanto, a probabilidade cumulativa que o raio do laser seja interrompido por n grades.

$$P(parada = verdade) = P_c(X, \boldsymbol{\rho}) = \sum_{i=1}^{k} P_c(x_i, \rho_i) \prod_{j=1}^{i-1} (1 - P_c(x_j, \rho_j)) \qquad (2\text{-}37)$$

onde $X = (x_1, x_2, ..., x_n)$ é o vetor de distâncias percorridas através das grades e $\boldsymbol{\rho} = (\rho_1, \rho_2, ..., \rho_n)$ são as opacidades correspondentes daquelas grades.

Assim, a probabilidade de que o raio do laser será interrompido em uma grade quadrada j é $P(parada = j)$, que é calculada como a probabilidade de que o laser atingiu o quadrado $j-1$ e, em seguida, parou em j, logo

$$P(parada = j) = P_c(x_j, \rho_j)(1 - P_c(X_{1:j-1}, \boldsymbol{\rho}_{1:j-1})) \qquad (2\text{-}38)$$

onde $X_{1:j-1}$ e $\boldsymbol{\rho}_{1:j-1}$ têm a interpretação natural como fragmentos dos vetores X e $\boldsymbol{\rho}$.

Suponha que $\boldsymbol{\delta} = (\delta_1, \delta_2, ..., \delta_n)$ é um vetor, tais que δ_i é a diferença entre o comprimento total da medição do laser e o comprimento até a grade i . Eliazar [21] define a probabilidade condicional de medição, dado que o raio do laser foi interrompido no quadrado i, como: $P_L(\delta_i | parada = i)$, para o qual a literatura faz uma suposição típica, de medição do ruído normalmente distribuído [34]. Observe que os termos δ_i são definidos apenas se a medição do laser observa um ponto específico de parada.

O evento $(\delta_i, parada = i)$ forma uma partição de todo possível evento de parada do laser. Assim, a probabilidade de medição segundo o teorema da probabilidade total é a soma, sobre todas as quadrículas do alcance do laser, do produto da probabilidade condicional da medição dado que o raio tenha parado, e a probabilidade de que o raio parou em cada grade.

$$P(L, parado = verdade) = \sum_{i=1}^{n} P_L(\delta_i | parada = i) P(parada = i) \qquad (2\text{-}39)$$

2.7.3 Atualização do Mapa

A média da distribuição exponencial com opacidade ρ é simplesmente ρ, o que faz a atualização do mapa particularmente simples. Para cada grade, basta manter a soma das distâncias totais d_t viajadas por um laser da varredura através da grade. Também é preciso manter registro do número de vezes que se presume que o laser tenha parado na grade h. Segundo [21], a estimativa de ρ é representada por:

$$\hat{\rho} = \frac{d_t}{h} \tag{2-40}$$

Grades que não tenham sido observadas por qualquer antepassado de uma partícula em consideração são tratadas de uma maneira especial, porque é difícil de estimar a probabilidade de parada do laser da Equaçaõ (2-38) sem experiência prévia. Eliazar propõe em [21] inicializar a probabilidade das grades desconhecidas como:

$$P_c(x,\rho) = 0 \tag{2-41}$$

O traço da linha do laser é continuado após o ponto observado até uma distância de 6σ, onde σ^2 é a variância do modelo de ruído do LRF. Isto permite ter a certeza de que qualquer ponto razoável na trajetória do laser é permitido ser uma possível fonte de observação [21]. Nos casos em que o laser viajar através quadrículas anteriormente não observadas, a probabilidade total calculada de que o laser será interrompido, dada a trajetória e o mapa, pode ser inferior a um. Neste caso, cada área do mapa desconhecida é dado o benefício da dúvida.

2.7.4 PD-Mapping

A maior contribuição da DP-SLAM é uma representação eficiente do mapa, ao mesmo tempo reduzindo a memória total exigida para representar um grande número de grades de ocupação. Isto é conseguido através de um método chamado de mapeamento de partículas distribuídas (PD-Mapping), o qual explora as redundâncias significativas entre os diferentes mapas.

Cada mapa possui muito em comum com a maioria dos outros mapas, e reproduzindo toda essa informação várias vezes é um uso ineficiente dos recursos de memória. Para tornar este conceito mais claro, vamos introduzir a noção de descendência de partículas.

Partícula Pai É uma partícula amostrada a partir da distribuição no tempo $t-1$, que produz partículas novas no tempo t.

Partículas Filhos São a geração de partículas no tempo t, que foram produzidas por uma partícula pai no tempo $t-1$

Partículas Irmãs São duas partículas com o mesmo pai.

Para ver como isso pode ser útil, suponha que um raio laser do LRF varre uma área de tamanho $A << M$ (onde M é o tamanho do mapa), e considere dois irmãos S_1 e S_2. Cada irmão corresponderá a uma posição diferente do robô e cada um fará atualizações ao mapa que herda de seu pai, máximo uma área do tamanho A. Assim, os mapas para S_1 e S_2 podem diferir do seu pai no máximo

em uma área de tamanho A. Todo o restante do mapa será idêntico.

Para que um raio do laser procure um obstáculo no mapa, seria necessário trabalhar com toda a descendência da partícula atual, e consultar uma lista de diferenças armazenada para cada partícula. A complexidade desta operação seria linear ao número de iterações do filtro de partículas. O desafio é, por conseguinte, fornecer as estruturas de dados que permitem atualizações eficientes para o mapa de localização e consultas eficientes com complexidade de tempo que seja independente do número de iterações do filtro de partículas. O conceito básico da árvore de descendência de partículas é detalhado em [4] e [2].

2.7.5 SLAM usando PD-Map

Acessando uma grade particular em um PD-map é um pouco mais complicado do que em uma grade de ocupação padrão. Uma vez que cada grade quadrada contém todo um conjunto de observações, descobrir se uma partícula específica fez uma observação neste local requer uma busca através deste conjunto de observações. No entanto, cada partícula herda as atualizações do mapa inserido por seus antepassados. Por isso, precisamos dar um passo para trás através da descendência da partícula, comparando cada ancestral com o conjunto de observações a este quadrado de grade.

Fazer uma atualização no mapa é um processo mais simples do que acessá-lo. Quando uma nova observação é feita, o valor atualizado é adicionado ao conjunto de observações do quadrado de grade. Ao mesmo tempo, a partícula ancestral que causou a atualização mantém um ponteiro para a observação específica no mapa.

Exclusões de entradas para o mapa são realizadas apenas quando uma partícula antepassada é removida. Usando a lista de atualizações de mapas para esta partícula ancestral, que foi criada quando as atualizações de mapas foram inicialmente adicionadas, podemos facilmente remover todas as observações pertinentes.

No próximo capítulo, os detalhes da implementação dos algoritmos no sistema experimental são apresentados.