

Análise do artigo “A Probabilistic Approach to Collaborative Multi-Robot Localization”

Nicolau Werneck

29 de Maio de 2008

1 Introdução e trabalhos anteriores

1.1 Robótica Móvel e localização compensatória

A robótica móvel é uma área de estudos peculiar por unir outras áreas mais gerais, como o projeto do equipamentos mecânicos, modelagem de sistemas, controle convencional e adaptativo, aprendizagem computacional, visão computacional, sistemas multi-agentes e outras. Seu objeto principal de estudo são robôs constituídos por sistemas de locomoção, controle, sensoriamento, recursos computacionais e às vezes também de comunicação. O objetivo é tornar os robôs cada vez mais autônomos e capazes de realizar tarefas mais complicadas.

As maiores dificuldades nos problemas da robótica móvel brotam em parte da necessidade de se integrar dados de múltiplos sensores, e também das heterogeneidades e incertezas nestas leituras e nos resultados das ações comandadas ao robô. A existência de incertezas levou com naturalidade à aplicação da teoria de probabilidades ao problema. Um bom exemplo disto é a formalização dos algoritmos de mapeamento que utilizam grades de ocupação [Mor88].

Os primeiros modelos probabilísticos foram utilizados para a integração de leituras de sensores incertos e heterogêneos. Estas técnicas foram aplicadas, por exemplo, no mapeamento detalhado de ambientes e na obtenção de dados de boa qualidade para serem utilizadas por sistemas de controle voltados a ações táticas de locomoção, como desviar de obstáculos. Os métodos de localização utilizados nesta época eram menos sofisticados do que o que apresentaremos a seguir.

Em geral a localização se resumia ou à detecção externa da posição do robô, ou dados obtidos por *dead-reckoning*, ou ainda o uso de técnicas estocásticas mais restritas, como o filtro de Kálmán, que atualizam iterativamente uma estimativa simplificada da posição do robô a partir de leituras sensoriais e modelos do ambiente e das ações. Era desejável aperfeiçoar estas técnicas para permitir maior autonomia, melhor aproveitamento do conhecimento do ambiente, melhor integração com informações tiradas de sensores ruidosos, e principalmente representações mais flexíveis das possíveis localizações do robô.

1.2 POMDP

Em meados de 1995 diversos grupos de pesquisa avançaram significativamente na questão da localização ao utilizarem o conceito de POMDP [NPB95; SK95; CKK96; BFHS96]. Este formalismo permitiu flexibilizar a representação de incertezas de localização, e também foi aproveitado inicialmente para a criação de modelos de localização mais abstratos. Ele ainda facilita muito a integração de leituras de sensores ruidosos e heterogêneos, e se baseia em teorias sólidas.

POMDP é a sigla para *partially observable Markov decision process*. Em um processo de decisão markoviano existe um conjunto de estados \mathbf{S} em que o sistema pode se encontrar, e um conjunto de ações \mathbf{A} que podem ser realizadas por uma entidade. No caso da robótica móvel os estados podem representar simplesmente a posição do robô, e as ações os comandos de deslocamento. Um POMDP também possui uma definição das probabilidades de ocorrer uma transição entre os estados a cada iteração do processo [Ros92].

As probabilidades de transição dependem dos estados do sistema e das ações selecionadas em cada iteração. A condição de Markov do processo significa que este “não possui memória”, ou seja, as probabilidades de transição não dependem dos estados e ações passadas se condicionadas ao presente:

$$s_{t+1} \perp\!\!\!\perp s_k, a_k | s_t, a_t, \quad k < t \quad (1)$$

Assim podemos definir o processo através de uma tabela das probabilidades $T(s, a, s')$ do sistema passar ao estado s' quando for selecionada a ação a e o sistema se encontrar no estado s .

O sistema é dito parcialmente observável porque a entidade que seleciona as ações não pode conhecer exatamente o estado em que o sistema se encontra, mas apenas o valor de variáveis que dependem probabilisticamente deste estado real, e que podem ser utilizadas para estimá-lo indiretamente.

O robô DERVISH [NPB95], vencedor de um concurso de robôs em ambiente de escritório em 1994, foi talvez o primeiro a aproveitar a idéia de representar a posição do robô com uma distribuição de probabilidades genérica sobre estados possíveis. Estes estados são nodos em um grafo representando porções do ambiente navegável — salas e corredores. No DERVISH, ações de locomoção e leitura de sensores modificam iterativamente a distribuição de probabilidades.

É interessante contrastar o DERVISH com o RHINO [BBC⁺95], por exemplo, que também participou do mesmo concurso. O RHINO também representa o espaço com um grafo, mas sua localização é feita apenas através de *dead-reckoning* com correções paulatinas, e não com funções de distribuição de probabilidade potencialmente genéricas.

A representação probabilística da localização do robô em um modelo topológico do espaço foi apresentada com muito mais rigor em artigos subsequentes de outros autores [SK95; CKK96], mas estes todos mantiveram o modelo topológico, apenas considerando a possibilidade de se associar mais informações métricas a cada nodo para permitir eventualmente uma localização mais precisa, mas subordinada hierarquicamente à localização mais abstrata, no grafo. Simmons chega a mencionar a possibilidade de se repartir o espaço em células

pequenas, mas apenas considera que o custo computacional deve impedir o uso de altas resoluções, e se limita a utilizar uma grade de 1 metro.

Apesar destas técnicas de localização utilizarem apenas modelos topológicos, ou com grades de baixa resolução, já haviam outras técnicas não-iterativas baseadas em representações “métricas”, e o mais importante, pesquisas que envolviam grades de ocupação de alta resolução [Mor88; ME85] para modelar o ambiente e lidar com as leituras dos sensores.

1.3 Localização de Markov e de Monte Carlo

Como já mencionamos, até meados de 1995 a localização de robôs era feita basicamente através de *dead-reckoning*, o que está fadado ao acúmulo de erros. Alguns métodos buscam apenas compensar estes erros, como o filtro de Kálmán. Outros ainda melhoram o cenário, permitindo PDFs flexíveis, mas apenas localizando o robô em um vértice de uma representação abstrata, topológica do ambiente. Este modelo de localização foi depois aperfeiçoado, transformado explicitamente em um problema de POMDP definido sobre rigorosos conceitos de teoria probabilística, e com uso da regra de Bayes para a assimilação de informações obtidas por sensores.

Ainda era desejável, entretanto, alguma forma de localização “métrica”, precisa porém aplicável no problema da localização global. Havia propostas para a comparação de leituras de sensores com modelos do ambiente, visando identificar as posições que poderiam causar as leituras em um dado momento [Elf87; Mil85]. Foi unindo estas técnicas às com mapas topológicos que Burgard *et alii* [BFHS96] criaram a técnica de localização baseada em uma *grade de probabilidade de posição*. Burgard não só explicitamente relaciona seu método a estes outros dois tipos, como também menciona se tratar de algo correspondente à grade de probabilidade de ocupação de Moravec.

Uma grade de probabilidade de localização modela a função de densidade de probabilidade (PDF) da posição do robô com uma discretização sobre uma grade regular. Em posse desta função é possível, por exemplo, descobrir o ponto de maior probabilidade, estimar a certeza da localização, ou manter hipóteses paralelas (e.g. devido a simetrias no ambiente).

Este algoritmo baseado em grade de probabilidade de localização, com atualizações a partir da verossimilhança dos sensores e modelos de transição, acabou recebendo o nome simplificado de “Markov localization” [Fox98; FBT99]. Uma melhoria subsequente foi a aplicação de uma técnica desenvolvida em outras áreas de estudo: A PDF passou a ser representada de forma estocástica, através de nuvens de amostras extraídas da função ao invés de medições precisas do valor por todo espaço. Esta forma do algoritmo recebeu por sua vez o nome simplificado de “Monte Carlo Localization” (MCL), e possui um excelente desempenho com boa precisão [FBTD99]. Seu funcionamento será explicado na próxima seção.

1.4 Localização Monte Carlo colaborativa

O artigo *A Probabilistic Approach to Collaborative Multi-Robot Localization* de Fox *et alii* [FBKT00] apresenta uma extensão ao Markov localization com múltiplos robôs em sinergia. Além do funcionamento normal, quando um robô detecta outro ele pode transmitir para o detectado informações para que este aperfeiçoe sua estimativa de localização, como qualquer outra informação sensorial. Esta detecção é feita com uma câmera e uma trena laser.

Um bom exemplo de funcionamento desta localização cooperativa, descrito no artigo, é quando temos um robô certo de sua posição, observando outro se deslocando perpendicularmente à sua linha de visão. A detecção reduz a incerteza do segundo robô, cuja PDF se concentra sobre um segmento de reta.

2 Detalhamento técnico

2.1 Localização individual

O algoritmo de localização de Markov funciona da seguinte forma: existe uma matriz contendo os valores da PDF no espaço, ou a função “crença” do robô $Bel^{(t)}(L = l)$ sobre a sua posição no espaço L no instante t . Esta matriz é atualizada a cada iteração. A necessidade de se armazenar apenas a crença, atualizando-a passo-a-passo, evidencia a dependência na condição de Markov. As atualizações pode ser das seguintes formas:

- *Atualização por movimentação*: Como a crença é a probabilidade das posições condicionada aos dados recebidos até o momento, temos em t

$$Bel^{(t)}(L = l) = P(L^{(t)} = l | d^{(t)}) \quad (2)$$

Considerando que $d^{(t)} = o^{(t)} \cup d^{(t-1)}$, temos pela regra de Bayes:

$$P(L^{(t)} = l | o^{(t)}, d^{(t-1)}) = \frac{P(o^{(t)} | L^{(t)} = l, d^{(t-1)}) P(L^{(t)} = l | d^{(t-1)})}{P(o^{(t)} | d^{(t-1)})} \quad (3)$$

Considerando o modelo do sensor dependendo apenas da posição, podemos simplificar para:

$$Bel^{(t)}(L = l) = \frac{P(o^{(t)} | L^{(t)} = l) P(L^{(t)} = l | d^{(t-1)})}{P(o^{(t)} | d^{(t-1)})} \quad (4)$$

Eliminado então o denominador (que pode ser obtido por normalização), e depois considerando que $d^{(t-1)}$ só permite inferir até $L^{(t-1)}$, temos

$$\begin{aligned} Bel^{(t)}(L = l) &= \alpha P(o^{(t)} | L^{(t)} = l) P(L^{(t)} = l | d^{(t-1)}) \\ &= \alpha P(o^{(t)} | L^{(t)} = l) P(L^{(t-1)} = l | d^{(t-1)}) \\ &= \alpha P(o^{(t)} | L^{(t)} = l) Bel^{(t-1)}(L = l). \end{aligned} \quad (5)$$

Ou seja, para considerar uma nova percepção em um dado momento basta modular a atual função de crença pela função de verossimilhança do sensor com o valor obtido dos sensores. Colocando de uma forma mais algorítmica, dado um o executamos sob um laço em l :

$$Bel(l) \leftarrow \alpha P(o|l) Bel(l). \quad (6)$$

- *Atualização por movimentação*: Neste caso temos $d^{(t)} = a^{(t)} \cup d^{(t-1)}$. Aqui utilizamos um conceito trazido de MDPs. A PDF nos estados é atualizada de acordo com a função de probabilidades de transição de cada estado original em função da ação, calculando-se seu valor esperado:

$$\begin{aligned} Bel^{(t)}(L = l) &= \int P(L^{(t)} = l | L^{(t-1)} = m, d^{(t)}) \cdot P(L^{(t-1)} = m | d^{(t)}) dm \\ &= \int P(L^{(t)} = l | L^{(t-1)} = m, a^{(t)}) \cdot P(L^{(t-1)} = m | d^{(t-1)}) dm \\ Bel^{(t)}(L = l) &= \int P(L^{(t)} = l | L^{(t-1)} = m, a^{(t)}) \cdot Bel^{(t-1)}(L = m) dm. \end{aligned} \quad (7)$$

Ou seja, dada uma ação a no instante t , atualizar a crença segundo

$$Bel(l) \leftarrow \int P(l|m, a) \cdot Bel(m) dm. \quad (8)$$

2.2 Localização colaborativa

No caso de localização colaborativa, cada robô mantém uma crença própria de sua posição. Quando necessário, diferenciaremos estas funções com um índice, como em $Bel_n^{(t)}(L = l)$. As atualizações individuais funcionam como explicado anteriormente. Entretanto, existe ainda o caso de atualização devido à percepção da posição de um robô relativa a outro. O artigo não apresenta outra atualização que seria possível: de uma detecção negativa, onde considera-se que um certo robô não está sendo capaz de observar o outro.

Esta atualização por detecção positiva funciona portanto de forma similar à atualização devido aos sensores, já que há uma modulação da função de crença por outra função. A diferença é que esta função é determinada a partir do valor esperado de uma função que modela o erro na medição da posição do robô detectado, de forma similar ao que ocorre na atualização devido a movimentação. Esta atualização, na forma algorítmica, é

$$Bel_n(l) \leftarrow Bel_n(l) \int P(l|m, r_k) \cdot Bel_k(m) dm, \quad (9)$$

onde o índice k representa o robô que realizou a detecção, n o detectado, e r_k os dados de sensores relativos a este evento. Uma transmissão de dados do robô que realizou a detecção para o detectado é necessária, visto que cada robô armazena apenas sua própria função crença, e esta atualização da função do robô detectado depende da função do outro.

2.3 Uso de técnicas Monte Carlo

O algoritmo proposto pode ser implementado utilizando-se técnicas de Monte Carlo, onde existe uma representação simplificada das funções de densidade de probabilidade. Esta técnica pode ser considerada menos rigorosa, e introduz no próprio funcionamento do sistema. Por outro lado o algoritmo adquire certas vantagens, como um custo menor e proporcional às necessidades apresentadas — quanto maior a certeza na localização, mais rápida a execução, e ainda a capacidade de poder ter a execução abortada a qualquer momento.

Com esta técnica representa-se uma distribuição por um conjunto de pontos sorteados dela. O número de pontos utilizados na amostragem é uma das preocupações naturais ao uso desta técnica. Estes pontos ainda são ponderados, de forma que o valor da densidade de probabilidade representada por eles depende tanto da densidade local de pontos, quanto de seus valores relativos.

É fácil ver as vantagens desta técnica. Primeiro, quando modulamos uma PDF, e.g. por uma verossimilhança, podemos simplesmente ponderar os pesos das amostras dadas. Modificações da função, como uma transformação afim do domínio, podem ser realizadas operando-se sobre as coordenadas de cada amostra. Por último, podemos implementar transformações lineares do tipo $g(y) = \int h(y|x)f(x)dx$, onde $f(x)$ é a PDF modelada e $h(y|x)$ uma outra PDF conhecida em qualquer ponto. Para isto realizam-se sorteios entre as amostras, ponderadas pelos pesos, e para cada x sorteado gera-se uma amostra de $h(y|x)$. Estas amostras acumuladas formam o modelo da nova PDF.

Amostras em pontos aleatórios são introduzidas paralelamente durante a operação do algoritmo para evitar mínimos locais. Se o robô cair numa situação em que a incerteza sobre sua posição começa a crescer, a crença começa a se aproximar de uma distribuição uniforme.

Até aqui vimos o uso da técnica conforme se observa em localização individual de robôs. Porém, um dos problemas gerados pelo uso desta representação da PDF surge justamente no momento de aproveitar a informação da detecção positiva. Quando um robô faz uma detecção, ele calcula a seguir, a partir de sua crença e do modelo da medição da posição relativa do outro robô, um conjunto de amostras modelando uma função no espaço que deve ser utilizada para modular a crença do robô detectado, atualizando-a.

Para permitir esta modulação (multiplicação) e também facilitar a transmissão da informação, cria-se um modelo de função (de domínio contínuo e constante-por-partes) a partir da nuvem de pontos [KF98; MSD97]. O artigo propõe uma k -d-tree montada por sucessivas bipartições das folhas obedecendo a um critério que depende das localizações e pesos das amostras. O valor da função em um certo ponto é proporcional à razão entre o peso das amostras na região, e a seu hiper-volume.

2.4 Algoritmo completo

O sistema para controle do robô n é constituído portanto por uma função crença $Bel_n(l)$ que é atualizada conforme diferentes tipos de informação se en-

contram à disposição. Podemos descrever o controlador da seguinte forma:

Algorithm 1 Controlador para MCL cooperativa

```

procedure ATUALIZA( $d$ ) ▷ Procedimento executado por interrupção
  if  $d$  é uma observação  $o$  then
    for all  $l$  do  $Bel(l) \leftarrow \alpha P(o|l) Bel(l)$ 
  else if  $d$  é uma ação  $a$  then
    for all  $l$  do  $Bel(l) \leftarrow \int P(l|m, a) \cdot Bel(m) dm$ 
  else if  $d$  é um aviso de detecção externa contendo  $Func(l)$  then
    for all  $l$  do  $Bel(l) \leftarrow Bel(l)Func(l)$ 
  else if  $d$  é uma detecção positiva  $r$  do robô  $q$  then
    for all  $l$  do
       $Func\_MC(l) \leftarrow \int P(l|m, r) \cdot Bel(m) dm$ 
       $Func(l) \leftarrow \text{CRIA\_KD-TREE}(Func\_MC(l))$ 
      TRANSMITE(para robô  $q$ ,  $Func(l)$ )

```

Onde as funções de probabilidade $P(\dots)$ possuem modelos predeterminados, com valores conhecidos em qualquer ponto. As funções Bel e $Func_MC$ são do tipo Monte Carlo, modeladas por conjuntos de amostras ponderadas, e $Func$ é modelada por uma kd -tree.

3 Conclusões

O artigo de For *et alii* apresenta ainda discussões sobre o sistema de identificação de robôs, baseado em uma câmera e um sensor laser, e demonstra testes realizados com o algoritmo. São apresentados um exemplo de aplicação em robôs reais, e simulações que mostram como a técnica permite integrar com facilidade robôs com sensores heterogêneos, permitindo que todos se beneficiem da diversidade.

Comparações da técnica com grupos de robôs tentando se localizar sem cooperação demonstra que este fator pode ser bastante relevante à eficiência, não só reduzindo o tempo com que uma boa estimativa da sua posição é encontrada por cada robô, mas ainda reduzindo o erro final.

Os autores reforçam algumas das limitações do algoritmo: Não foi explorada a possibilidade de incorporar detecções negativas, e é preciso que a detecção e identificação dos outros robôs se dê de forma bastante confiável. Além de atacar estas questões, algumas das extensões mais interessantes previstas seriam a criação de uma política de controle para localização ativa, e um desenvolvimento do controle de incorporações de informações à crença, levando em consideração as condições em que os robôs se apresentam.

Referências

[BBC⁺95] Joachim M. Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank E. Schneider, Jiannis Strikos, and Se-

- bastian Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, 1995.
- [BFHS96] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *AAAI/IAAI, Vol. 2*, pages 896–901, 1996.
- [CKK96] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
- [Elf87] A. Elfes. Sonar-based real-world mapping and navigation. *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, 3(3):249–265, Jun 1987.
- [FBDT99] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *AAAI/IAAI*, pages 343–349, 1999.
- [FBKT00] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.
- [FBT99] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
- [Fox98] Dieter Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. PhD thesis, University of Bonn, Germany, 1998.
- [KF98] Daphne Koller and Raya Fratkina. Using learning for approximation in stochastic processes. In *Proc. 15th International Conf. on Machine Learning*, pages 287–295. Morgan Kaufmann, San Francisco, CA, 1998.
- [ME85] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2:116–121, Mar 1985.
- [Mil85] D. Miller. A spatial representation system for mobile robots. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2:122–127, Mar 1985.
- [Mor88] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Mag.*, 9(2):61–74, 1988.

- [MSD97] Andrew W. Moore, Jeff Schneider, and Kan Deng. Efficient locally weighted polynomial regression predictions. In *Proc. 14th International Conference on Machine Learning*, pages 236–244. Morgan Kaufmann, 1997.
- [NPB95] Illah R. Nourbakhsh, Rob Powers, and Stan Birchfield. Dervish - an office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.
- [Ros92] Sheldon M. Ross. *Applied Probability Models with Optimization Applications*. Dover Publications, 1970 edition edition, 1992.
- [SK95] Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1087, 1995.