

MISTURA DE GAUSSIANAS: UMA ABORDAGEM RÁPIDA PARA MODELAR NUVEM DE PONTOS

SIDNEI DA SILVA FILHO*, PAULO DREWS-JR*, LUIZ F. V. MARCOLINO*

**Grupo de Automação e Robótica Inteligentes (NAUTEC),
Centro de Ciências Computacionais, Universidade Federal do Rio Grande, RS, Brasil*

Emails: `sidneifilho@furg.br`, `paulodrews@furg.br`, `luiz.fernando@furg.br`

Abstract— This paper proposes an adaptable system to build a 3D feature model of point clouds using Gaussian Mixture Models. Nowadays, scanners are capable of acquire large point cloud with a good benefit-cost ratio regarding the acquiring time, accuracy and price. However, dealing with this data is a hard task due the amount of points, the redundancy data and noise. Therefore, this work presents an efficient method based on top-down approach to estimate features in 3D data. It aims at dealing with local minimum and initialization problem of the Gaussian Mixture Models estimation based on the classic EM algorithm. This paper also compares two selection criteria. The results show that system is able to estimate Gaussian Mixture Models in 3D data with limited computational cost.

Resumo— Este artigo propõe o uso de mistura de Gaussianas como um sistema adaptável para construção de um modelo nuvens de pontos tridimensionais. Hoje em dia, scanners são capazes de adquirir grandes nuvens de pontos com uma boa relação custo-benefício em relação ao tempo de aquisição e precisão. Contudo, lidar com esses dados é uma tarefa difícil devido a quantidade de pontos, redundância de dados e ruídos. Portanto, este trabalho apresenta um método eficiente baseado em uma abordagem top-down para representar nuvens de pontos 3D, visando ultrapassar restrições de inicialização e ótimos locais do algoritmo Expectation-Maximization. Além disso, este trabalho compara dois critérios de seleção. Os resultados mostram que o sistema proposto é capaz de estimar uma boa mistura de Gaussianas com reduzido custo computacional.

Palavras-chave— Misturas de Gaussianas, Nuvem de Pontos 3D, Expectation Maximization

1 Introdução

O modelagem de nuvens de pontos é importante em diversos contextos, entre eles podemos destacar o uso na robótica. É importante obter uma representação de alto nível dos dados devido ao tamanho típico das nuvens de pontos e restrições de tempo. Assim, uma abordagem que permite lidar com nuvens de pontos em alto nível pode reduzir a exigência em aplicações críticas, tais como detecção de lugares (Swadzba and Wachsmuth, 2011; Shi et al., 2011), reconhecimento de objetos (Faria et al., 2009; Rusu et al., 2009) e detecção de mudanças (Vieira et al., 2012; Drews-Jr et al., 2013).

Nos campos de processamento de sinais e análise de dados, as misturas de Gaussianas são largamente estudadas e aplicadas (Bishop, 2007). Portanto, podem ser usadas para modelar nuvens de pontos geométricas tridimensionais, aonde são capazes de fornecer um modelo compacto (Drews-Jr et al., 2010). Contudo, a principal limitação dessa abordagem é o custo computacional para se obter uma boa mistura, como mostrado em (Núñez et al., 2010).

Quando o número de Gaussianas da mistura é conhecido, o custo computacional envolvido é reduzido. Nesse caso, o algoritmo *Expectation Maximization* (EM) pode ser aplicado para estimar os parâmetros da mistura (Dempster et al., 1977). Ele tipicamente converge para modelos representativos, mesmo com uma inicialização aleatória. No entanto, a convergência só é garantida para

um ótimo local.

Quando as nuvens de pontos são adquiridas por sensores, tais como *laser scanner*, câmeras de par estéreo ou sensores de profundidade, não há conhecimento a respeito do número de formas a serem modeladas, até mesmo uma estimativa é impossível de se obter. Por isso, o número de componentes deve ser estimado automaticamente. No trabalho de Drews *et al.* (Drews-Jr et al., 2013), é apresentado um algoritmo baseado no paradigma *split-and-merge* que determina o número de componentes da mistura automaticamente, além de superar as restrições de mínimos locais.

Uma grande limitação desses algoritmos é o critério de parada. De modo a lidar com esse problema, o presente trabalho compara dois critérios de seleção com características distintas. No artigo de (Drews-Jr et al., 2013) foram comparados critérios de seleção que são denominados funções de penalização. Dentre esses critérios o que obteve a melhor mistura de Gaussianas com o menor tempo de processamento foi o *Minimum Description Length* (MDL) proposto em (Figueiredo and Jain, 2002). O presente trabalho pretende comparar o MDL com outro tipo de critério de seleção. O critério escolhido é o *BYY-Harmony Data Smoothing* (BYY-HDS) (Xu, 2002). Este critério foi criado com o intuito de obter mínimos globais e determinar o número correto de componentes da mistura.

Este artigo está organizado da seguinte forma. Após analisar brevemente o estado da arte na Seção 2, a Seção 3 apresenta uma revisão sobre mis-

tura de Gaussianas e o algoritmo EM. Na Seção 4 é apresentado o algoritmo proposto para estimar as componentes da mistura. Os resultados são descritos na Seção 5. Finalmente, na Seção 6, as principais conclusões e trabalhos futuros são traçados.

2 Trabalhos Relacionados

O problema de modelar nuvem de pontos é tratado em diferentes áreas. Uma das alternativas de modelagem é construindo superfícies implícitas (Vieira et al., 2012). Embora seja um método eficiente, quando a superfície a ser modelada não é fechada, o método pode encontrar dificuldade de representação.

O uso de formas básicas também pode ser uma possibilidade de representação, embora o custo de estimação seja elevado, sendo assim restrito como mostrado em (Drews-Jr et al., 2010).

Nesse contexto, agrupamentos utilizando mistura de Gaussianas surgem como uma abordagem promissora (Núñez et al., 2010). Embora as maiores restrições estejam relacionadas com o tempo de processamento e também com a necessidade de conhecer previamente o número de Gaussianas presentes no conjunto de dados, o que não é conhecido.

Nesse caso, algumas abordagens como o trabalho de (Figueiredo and Jain, 2002) propõem metodologias iterativas que são muito custosas para grandes volumes de dados, caso das nuvem de pontos tridimensionais. Assim, o presente trabalho busca outras alternativas que utilizam o EM para estimar a mistura, bem como um bom critério de seleção.

3 Estimando Mistura de Gaussianas

Uma breve revisão sobre mistura de Gaussianas e sobre o algoritmo EM será feita.

3.1 Mistura de Gaussianas

Dado uma variável aleatório X de dimensão d e uma mistura com K componentes. A função de probabilidade de mistura de Gaussianas pode ser definida por:

$$\Phi(X|\Theta_k) = \sum_{i=1}^K \pi_i \phi(X|\theta_i), \quad (1)$$

onde cada θ_i corresponde ao conjunto de parâmetros definidos pela i -ésima componente da mistura, $\pi_i \in [0, 1]$ com $i \in (1, 2, \dots, K)$ e $\sum_{i=1}^K \pi_i = 1$. O vetor $\Theta_k = (\pi_1, \dots, \pi_k, \theta_1, \dots, \theta_k)$ é o conjunto dos parâmetros da mistura. Cada componente $\phi(X|\theta_i)$ da mistura é uma função de densidade de

probabilidade Gaussiana definida por:

$$\phi(X = x|\theta_i) = \frac{1}{(2\pi^{\frac{d}{2}})|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)}, \quad (2)$$

onde μ_i é a média, Σ_i é a matriz de covariância e $\theta_i = (\mu_i, \Sigma_i)$ representa os parâmetros de uma Gaussiana.

Especificamente, dado um conjunto de dados S com N instâncias s_t e dimensão d , os parâmetros de Θ_k são estimados maximizando o logaritmo da seguinte função de verossimilhança:

$$\begin{aligned} L(\Theta_k|S) &= \log \prod_{t=1}^N \Phi(s_t|\Theta_k) \\ &= \sum_{t=1}^N \log \sum_{i=1}^K \pi_i \phi(s_t|\theta_i). \end{aligned} \quad (3)$$

3.2 Expectation Maximization

O EM (Dempster et al., 1977) é um algoritmo capaz de encontrar um ótimo local da função de máxima verossimilhança de uma mistura de Gaussianas. O EM possui duas etapas iterativas:

- **E-Step:** Nessa etapa ele calcula a probabilidade de cada ponto pertencer a cada Gaussiana. Além disso, é calculada uma nova estimativa da função de verossimilhança utilizando a equação 4.

$$P(\pi_k, \theta_k | s_t) = \frac{\pi_k \phi(s_t, \theta_k)}{\sum_{t=1}^N \pi_t \phi(s_t, \theta_t)}. \quad (4)$$

- **M-Step:** Nessa etapa, as componentes da mistura são maximizadas através das seguintes equações:

$$\begin{aligned} \pi_k &= \frac{1}{N} \sum_{t=1}^N P(\pi_k, \theta_k | s_t), \\ \mu_k &= \frac{\sum_{t=1}^N P(\pi_k, \theta_k | s_t) s_t}{\sum_{t=1}^N P(\pi_k, \theta_k | s_t)}, \\ \Sigma_k &= \frac{\sum_{t=1}^N P(\pi_k, \theta_k | s_t) (s_t - \mu_k)(s_t - \mu_k)^t}{\sum_{t=1}^N P(\pi_k, \theta_k | s_t)}. \end{aligned} \quad (5)$$

4 Metodologia proposta

Este trabalho é baseado na modificação do algoritmo EM, onde o não se tem conhecimento a priori sobre o número de componentes K da mistura. A Figura 1 mostra uma visão global da abordagem proposta.

Inicialmente, na etapa do pré-processamento, a nuvem de pontos é simplificada e os ruídos são filtrados. Depois, o algoritmo *Split-EM* (Drews-Jr et al., 2013), é utilizado a fim de evitar mínimos locais e estimar uma representação adequada da

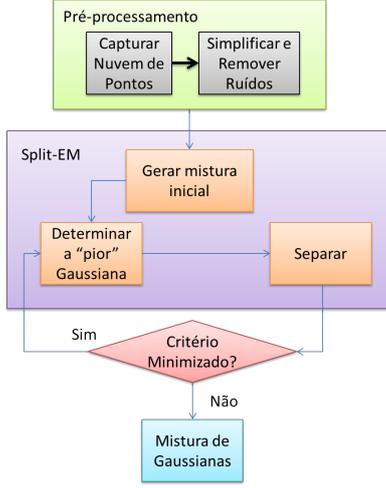


Figura 1: Visão Geral da Metodologia Proposta.

mistura de Gaussianas. Em seguida, é utilizado um critério de seleção para identificar se a mistura estimada é adequada.

4.1 Pré-processamento da nuvem de pontos 3D

O pré-processamento da nuvem de pontos é onde ocorre a simplificação da nuvem e a filtragem dos *outliers*.

O *Voxel Grid 3D* é um algoritmo clássico para simplificar uma nuvem de pontos de maneira eficiente. Ele divide a nuvem em voxels com tamanhos determinados. Depois, é calculado o centróide que representará os pontos que dentro de cada voxel.

O algoritmo *Statistical Outlier Removal* (Rusu et al., 2008) é responsável por reduzir *outliers*. Ele é baseado em uma distribuição normal da vizinhança de cada ponto, onde todos os pontos que têm sua distância fora do intervalo definido são removidos.

4.2 Critério de seleção

Critérios de seleção são utilizados para identificar um bom modelo. Para o caso de mistura de Gaussianas, o objetivo é minimizar a seguinte equação:

$$\hat{\Theta}_k = \operatorname{argmin}(C(\Theta_k|S)), \quad (6)$$

onde $C(\Theta_k|S)$ é o resultado do critério de seleção e Θ_k é o modelo de mistura de Gaussianas com K componentes.

Neste trabalho foram avaliados dois critérios de seleção diferentes, explicados na seção a seguir.

4.2.1 Minimum Description Length

Este critério é definido por: $C(\Theta_k|S) = -L(\Theta_k|S) + P(k)$, onde S é o conjunto de dados, a nuvem de pontos no nosso caso, $P(k)$ é a função de penalização que aumenta à medida que o número de Gaussianas aumentam e o $L(\Theta_k|S)$

é a verossimilhança calculada com a Equação 3. Este critério é definido pela seguinte equação:

$$C_{MDL}(\Theta_k|S) = -L(\Theta_k|S) + \frac{d}{2} \sum_{i=1}^k \log\left(\frac{N\pi_i}{12}\right) - \frac{k}{2} \log\left(\frac{N}{2}\right) - \frac{k(d+1)}{2}. \quad (7)$$

4.2.2 BYY Harmony Data Smoothing Learning

O BYY-HDS é uma estrutura de aprendizagem estatística geral, a partir do qual foram extraídos vários critérios e métodos de seleção automático (Xu, 2002). Este critério é definido por:

$$C_{BYY-HDS}(\Theta_k, h|S) = \sum_{k=1}^K \pi_l (0.5 \ln |\Sigma_k| + 0.5 h^2 \operatorname{Tr}[\Sigma_l^{-1}] - \ln \pi_k), \quad (8)$$

onde h é o parâmetro de suavização, $\theta_k^h = [\theta_k, h]$ e θ_k^h são obtidos pela suavização dos dados, regulando a máxima verossimilhança. Este é estimado usando um algoritmo de suavização do EM que possui os seguintes passos:

Passo 1: Calcular a probabilidade posterior usando a equação (4);

Passo 2: Atualizar os parâmetros do modelo por meio das equações do EM (5). A matriz de covariância é modificada usando um novo parâmetro de suavização h :

$$\Sigma_k = \frac{\sum_{t=1}^N P(\pi_k, \theta_k | s_i) (s_t - \mu_k)(s_t - \mu_k)^t}{\sum_{t=1}^N P(\pi_k, \theta_k | s_i)} + h^2 I \quad (9)$$

onde I é a matrix identidade.

Passo 3: Atualizar o parâmetro de suavização "h":

$$h_{new} = h_{old} + \eta_0 g(h_{old}), \quad (10)$$

onde η_0 é um passo de comprimento constante,

$$g(h_{old}) = \frac{D}{h_{old}} - h_{old} \sum_{k=1}^K \pi_k \operatorname{Tr}[\Sigma_k^{-1}] - \frac{\sum_{t=1}^N \sum_{v=1}^N \gamma_{t,v} \|s_t - s_v\|^2}{h_{old}^3}. \quad (11)$$

e

$$\gamma_{t,v} = \frac{e^{-0.5 \frac{\|s_t - s_v\|^2}{h_{old}^2}}}{\sum_{t=1}^N \sum_{v=1}^N e^{-0.5 \frac{\|s_t - s_v\|^2}{h_{old}^2}}} \quad (12)$$

Nas equações acima, K é um número de parâmetros da mistura (Ververidis and Kotropoulos, 2005), N é o número de pontos da nuvem e $d = 3$ é a dimensão de s . Esses critérios de seleção são aplicados no algoritmo *Split-EM* como condição de parada.

4.3 Algoritmo Split-EM

O algoritmo *Split-EM* (Drews-Jr et al., 2013) estima a mistura Gaussianas da seguinte forma. A mistura inicialmente contém apenas uma Gaussiana, com $\pi = 1$, μ igual a média dos pontos da nuvem e Σ a matriz de covariância simétrica positiva das amostras. O algoritmo determina qual a Gaussiana que representa de uma forma menos adequada o seu conjunto de dados, em seguida separa a mesma em duas novas Gaussianas com o intuito de representar melhor esses dados. Para identificar essa Gaussiana utilizou-se a métrica de entropia. Dada uma variável aleatória X e uma componente de mistura, a entropia é definida por:

$$\begin{aligned} H^k(S) &= -E_s[\log(p(S))] \\ &= -\frac{1}{N} \sum_{t=1}^N \log \phi(s_t | \theta_k) \end{aligned} \quad (13)$$

onde N é o número de amostras que pertencem a distribuição e $\phi(S|\theta_k)$ é a função de densidade local da k -ésima componente da mistura.

Outro valor importante é a entropia máxima. Trata-se de um limite para o valor da entropia. Este valor pode ser calculado a partir da seguinte equação:

$$H_{max} = -\frac{1}{2} \log[(2\pi)^d |\Sigma|] \quad (14)$$

A Sp_k é a diferença entre $H^k(X)$ e $H_{max}^k(X)$. A Gaussiana da mistura com a menor taxa, é dividida em outras duas Gaussianas com $\pi_{split} = \frac{\pi_k}{2}$, as médias serão calculadas usando o algoritmo *K-Means* (Ueda and Nakano, 1998) e a matriz de covariância é determinada utilizando a equação abaixo

$$\Sigma_k = \frac{1}{N} \sum_{t=1}^N (s_t - \mu)(s_t - \mu)^t \quad (15)$$

O algoritmo principal tem a seguinte estrutura:

1. Inicialmente, a mistura de Gaussianas é composta por apenas uma componente. Em seguida, a distribuição é atualizada usando o algoritmo EM e o valor de C_{Θ_k} é computado.
2. **Split:** Dados K parâmetros Θ_k , a Gaussiana com menor Sp_k é dividida. O EM atualiza os parâmetros da divisão Θ_{split} , onde a nova mistura de Gaussianas tem $k + 1$ componentes. Além disso, $C_{\Theta_{split}}$ é computado.
3. Se $C_{\Theta_{split}|S} > C_{\Theta_k|S}$ então a nova mistura de Gaussianas é aceita, senão, retorna ao passo 2.



Figura 2: Plataforma de experimentos: robô Robex e o sistema de percepção 3D baseado no laser Hokuyo URG-30LX.

Em cada iteração, a mistura é atualizada e é identificada a Gaussiana a ser dividida. Finalmente, o critério de seleção é utilizado para saber se a nuvem de pontos está sendo bem representada pela mistura de Gaussianas.

5 Experimentos

Nesta seção, será analisado o desempenho e precisão do algoritmo *Split-EM* em experimentos com dados reais. O EM clássico irá ser utilizado no algoritmo que contém o MDL e o EM suavizado no algoritmo que utiliza o BYY-HDS. A estimativa da componente inicial foi obtida através da atribuição aleatória observada em subpopulações e computando a média e matriz de covariância. O parâmetro de suavização h usado no algoritmo contendo o BYY-HDS é inicializado como $h^2 = \frac{1}{DN^3} \sum_{t=1}^N \sum_{v=1}^N \|s_t - s_v\|^2$. O comprimento de cada voxel usado no algoritmo de simplificação é definido como $5cm \times 5cm \times 5cm$. O desvio padrão e o número de médias utilizado no algoritmo *Statistical Outliers Removal* são definidos como 1.0 e 80 (Rusu et al., 2008).

Os experimentos foram realizados em um PC com processador Intel Core i7 3.4HZ, 4Gb de DDR3 RAM e GNU-Linux Ubuntu 12.04. A plataforma robótica utilizada para adquirir as nuvens de pontos 3D é o robô Robex (Gutiérrez et al., 2011), Fig. 2. Este é um robô projetado no RoboLab, Universidade de Extremadura - Cáceres, Espanha. Para os experimentos descritos neste artigo, um sensor 3D foi montado na parte superior. É composto por um laser HOKUYO URG-30LX em uma unidade de *pan-tilt*.

5.1 Comparação entre os critérios de seleção

Foram capturadas quinze nuvens de pontos de ambientes reais com o robô Robex a fim de validar o algoritmo proposto. Nas Figuras 3 e 4 são mostrados os resultados de dois experimentos. Essas figuras contêm imagens do ambiente real, nuvens de pontos e misturas de Gaussianas estimadas com cada critérios de seleção. As Gaussianas são representadas pelos elipsóides coloridos para facilitar a

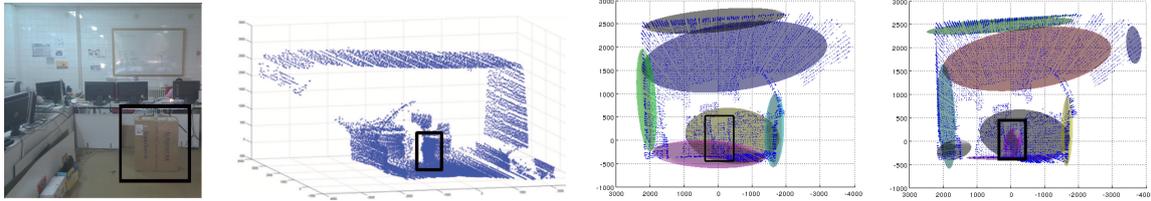


Figura 3: **Experimento A:** a) Foto do ambiente onde uma caixa foi inserida, b) Nuvem de pontos 3D capturada pelo robô Robex com mudança destacada em preto, c) Mistura de Gaussianas adquirida utilizando critério BYY-HDS contendo seis Gaussianas, d) Mistura de Gaussianas adquirida com o critério MDL contendo onze Gaussianas.

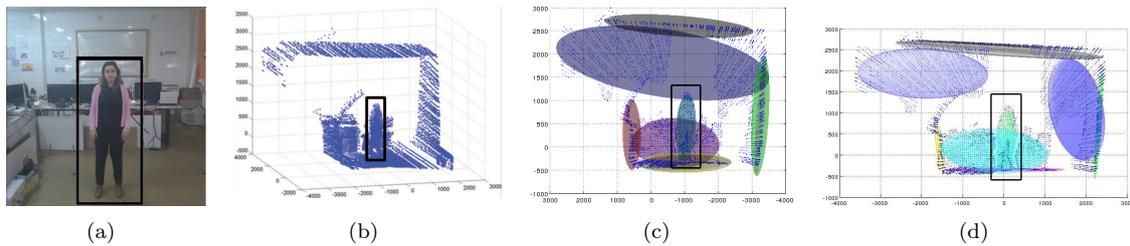


Figura 4: **Experimento B:** a) Foto do ambiente onde uma pessoa representa a mudança, b) Nuvem de pontos capturada pelo robô Robex com mudança destacada em preto, c) Mistura de Gaussianas adquirida pelo critério BYY-HDS contendo sete Gaussianas, d) Mistura de Gaussianas adquirida com critério MDL contendo nove Gaussianas.

visualização com $\sigma = 1.5$.

Ambos os experimentos foram realizados em ambiente *indoor*, como mostra as figuras 3(a) e 4(a). O experimento *A* apresenta como mudança a inserção de uma caixa. A nuvem de pontos adquirida pelo robô com a caixa é mostrada na Figura 3(b). Neste ambiente, a mistura de Gaussianas ótima deve conter sete componentes distintas, segundo estimativa manual. O critério de seleção BYY-HDS estimou uma mistura composta por seis Gaussianas, enquanto o MDL estimou uma mistura contendo onze Gaussianas como mostrado nas Figuras 3(c) e 3(d). Apesar de ambos os critérios não terem sido capazes de estimar o número correto de Gaussianas, o MDL mostrou-se mais eficaz do que o BYY-HDS, pois foi capaz de representar todos os objetos com forma distinta.

O experimento *B* foi realizado em um escritório onde uma pessoa é inserida na cena, como mostrado na Figura 4(a). A mistura de Gaussianas mais adequada para esse conjunto de dados é composta por 7 componentes. O algoritmo que utilizou o critério de seleção BYY-HDS encontrou uma mistura com 7 Gaussianas, enquanto que o algoritmo que utilizou o MDL determinou uma mistura com nove componentes. O BYY-HDS conseguiu estimar o número correto de Gaussianas, o MDL apesar de não determinar a melhor representação também conseguiu determinar uma mistura para a nuvem de pontos que segmentou todos objetos.

Como mostrado nas Figuras 3 e 4, existe uma diferença muito pequena entre os dois critérios.

Assim, resultados quantitativos foram obtidos a fim de definir o melhor critério. A Tabela 1 mostra os resultados de ambos os experimentos (*A* e *B*), considerando a segmentação dos objetos na cena e o tempo de execução médio do algoritmo utilizada cada critério distinto.

Os resultados da coluna *C* na Tabela 1 mostram que os dois métodos foram capazes de encontrar apenas 3 segmentações perfeitas (20%), considerando-se apenas o número de formas. Os resultados utilizando MDL são mais eficazes do que o critério de seleção BYY-HD. Na maioria dos experimentos o MDL conseguiu segmentar de forma distinta todos os objetos da cena, embora em alguns casos sobreestimando. Além disso, os resultados mostram que o tempo médio de execução utilizando o critério MDL é bem menor do que utilizando o BYY-HDS. Assim, o critério MDL apresentou melhores resultados que o BYY-HDS. Sendo capaz de estimar uma mistura de Gaussianas de maneira mais robusta e com menor custo computacional.

Tabela 1: Avaliação Quantitativa dos experimentos *A* e *B*, onde: Forma Subestimada (*U*), Adequada (*A*), Sobreestimada (*O*), Correta (*C*), Errada ($E=U+O$) e Tempo Médio (*s*).

Criteria/Exp.	U	A	O	C	W	Tempo
BYY-HDS	12	3	0	3	12	22.23
MDL	1	8	11	3	7	2.83

6 Conclusões e Trabalhos Futuros

Este trabalho avaliou uma metodologia para representação de nuvem de pontos utilizando mistura de Gaussianas. O algoritmo mostrou-se eficiente quando comparado ao estado-da-arte (Drews-Jr et al., 2010). Este sofre menos com problemas de mínimos locais como no EM clássico, além de ter um custo computacional menor.

Foram avaliados dois critérios de seleção: MDL e o BYY-HDS. Os experimentos mostraram que o MDL obteve melhores resultados, sendo capaz de determinar uma boa representação para a nuvem de pontos com menor custo computacional.

Os trabalhos futuros serão focados na melhoria do desempenho da estimação das misturas de Gaussianas utilizando processadores gráficos e uma nova versão em paralelo do algoritmo está em desenvolvimento. Além disso, pretende-se uma maior experimentação de modo a obter uma avaliação mais precisa.

Agradecimentos

Os autores gostariam de agradecer aos colegas do Robolab da Universidad de Extremadura, em especial ao Prof. Pedro Núñez, pela colaboração com os experimentos. Além disso, agradecer ao financiamento parcial da pesquisa dado pelo CNPQ, FAPERGS, FINEP e FURG.

Referências

- Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*, Springer.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*. **39**(1): 1–38.
- Drews-Jr, P., Filho, S., Marcolino, L. and Núñez, P. (2013). A fast and adaptive 3d change detection algorithm for autonomous robots based on Gaussian mixture models, *IEEE International Conference in Robotics and Automation*, pp. 4670 – 4675.
- Drews-Jr, P., Núñez, P., Rocha, R., Campos, M. and Dias, J. (2010). Novelty detection and 3D shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robot, *IEEE International Conference on Robotics and Automation*, pp. 3635–3640.
- Faria, D. R., Prado, J. A. S., Drews-Jr, P. and Dias, J. (2009). Object shape retrieval through grasp exploration, *European Conference on Mobile Robots - ECMR*.
- Figueiredo, M. and Jain, A. (2002). Unsupervised learning of finite mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(3): 381–396.
- Gutiérrez, M., Martineta, E., Sánchez, A., Rodríguez, R. and Núñez, P. (2011). A cost-efficient 3D sensing system for autonomous mobile robots, *XII Workshop of Physical Agents 2011 (WAF'2011)*.
- Núñez, P., Drews-Jr, P., Bandera, A., Rocha, R., Campos, M. and Dias, J. (2010). Change detection in 3D environments based on Gaussian mixture model and robust structural matching for autonomous robotic applications, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2633–2638.
- Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. and Beetz, M. (2008). Towards 3D point cloud based object maps for household environments, *Robotics and Autonomous Systems* **56**(11): 927–941.
- Rusu, R. B., Marton, Z. C., Blodow, N., Holzbach, A. and Beetz, M. (2009). Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments, *IEEE RSJ International Conference on Intelligent Robots and Systems*, pp. 3601 – 3608.
- Shi, L., Kodagoda, S. and Ranasinghe, R. (2011). Fast indoor scene classification using 3D point clouds, *Australasian Conference on Robotics and Automation*.
- Swadzba, A. and Wachsmuth, S. (2011). Indoor scene classification using combined 3d and gist features, *10th Asian conference on Computer vision - Volume Part II*.
- Ueda, N. and Nakano, R. (1998). Deterministic annealing em algorithm, *Neural Networks* **11**(2): 271 – 282.
- Ververidis, D. and Kotropoulos, C. (2005). Emotional speech classification using Gaussian mixture models, *IEEE International Symposium on Circuits and Systems*, Vol. 3, pp. 2871 – 2874.
- Vieira, A. W., Drews-Jr., P. and Campos, M. (2012). Efficient change detection in 3D environment for autonomous surveillancerobots based on implicit volume, *IEEE International Conference on Robotics and Automation*, pp. 2999 – 3004.
- Xu, L. (2002). BYY harmony learning, structural RPCL, and topological self-organizing on mixture models, *Neural Networks* **15**(8-9): 1125–1151.