Detecting Changes in 3D Maps using Gaussian distribution

Sidnei Silva Filho, Paulo Drews-Jr, Silvia Botelho  
Intelligent Robotics and Automation Group (NAUTEC),  
Computational Sciences Center, Universidade Federal do Rio Grande, Rio Grande, RS, Brazil  
E-mails: {sidneifilho, paulodrews, silviacb}@furg.br

Abstract—The improvement of technology allows robots to acquire dense point clouds decreasing the price and increasing the performance. However, it is a hard task to deal with it due to the large amount of points, the redundancy and the noise. This paper proposes methodologies in order to model 3D point clouds using Gaussian distribution. This model is used to detect changes in the autonomous robot’s working environment. A reference point cloud is modeled using the Split-EM algorithm [1] and use the Spatial Difference Probability metric [2] to change detection. The algorithm proposed by this work needs only the reference model differently from [1]. The results compare the proposed method with a state-of-art method in real scenarios, where the Split-EM presents high computational cost to model the point clouds but enable us to detect changes with a small cost. Furthermore, the proposed method does not depend on adjustment of parameters.

Keywords—Change Detection, Split-EM algorithm, Computer Vision, Gaussian Mixture Models.

I. INTRODUCTION

The change detection problem is important in several contexts, among them, we can highlight the use in robotics mapping, exploring tasks [3], during the navigation [4], [5] and loop-closure detection in Simultaneous Localization and Mapping methods (SLAM) [6], [7]. In the context of point clouds, it is important to obtain a high level representation due the typical scalability, time and computational power constraints.

In the field of signal processing and data analysis, the Gaussian Mixture Models (GMM) are widely studied and applied [8]. The GMM appears as an alternative modeling technique of 3D point clouds. And, it provides a compact probabilistic model [3]. However, the main limitation of this approach is the computational cost to obtain a GMM, as shown in [9]. This problem can be reduced if the number of the Gaussians in the mixture is known. Thus, the Expectation-Maximization (EM) algorithm can be used to estimate the GMMs parameters [10]. The convergence of the EM algorithm is only guaranteed to a representative model, even with a random initialization [11].

Point clouds are acquired by a large range of sensors commonly used in robotics as laser scanner, stereo cameras or depth sensor. In these data, the number of shapes to be modeled are unknown, even an estimate is impossible to obtain. So, the number of Gaussians should be automatically estimated. In the work of Drews et al. [1] was proposed an EM algorithm based on split paradigm to estimate the GMMs components using the function cost Minimum Description Length (MDL) [11]. It overcomes the local minimum constraints and the requirement to know a priori the number of components in the mixture. Even with these advantages, the method proposed is capable to model just a representative mixture for both reference and current 3D point clouds.

An alternative method called 3D-NDT algorithm [12] considers a set of points in the point clouds as a Gaussian distribution. In the case of change detection, it simplifies the modeling process, but the detection process is overloaded. The main advantage of this technique is switch the computational burden from the modeling process to the detection process. Andreasson et al. [2] proposed a methodology based on the 3D-NDT algorithm to detect changes in the robot’s working environment. The main idea is to model the reference data as a set of Gaussians and compare directly the current data with the mixture using the Spatial Difference Probability metric.

This work proposes a different method that take into account the advantage of both methods Drews et al. [1] and Andreasson et al. [2], principally in the context of change detection. The Split-EM algorithm finds the GMM to the reference 3D point cloud, and the current point cloud is compared with this model using the Spatial Difference Probability metric. Although, the high computational cost of the Split-EM algorithm, it generates a more compact and robust model that reduces the global computational cost.

This paper is organized as follows. The Section II briefly reviews the state of art. After, the Section III presents a review about Gaussian mixture models. Section IV shows the proposed methodology, and Section V describes the implemented 3D-NDT. Experimental results compare the proposed methodology with the 3D-NDT algorithm in Section VI. Finally, in Section VII, the main conclusions and future work are drawn.

II. RELATED WORKS

Nowadays, the research in autonomous robots working on dynamic environments has been increasing. In several approaches, the main strategy is to ignore the dynamic object from the modeling process with the purpose of improve the navigation and localization tasks. On the other hand, these changes in the robot’s surroundings may actually become relevant depending on the application. In this sense, some works addressed the problem of change detection. In the context of surveillance robots, the problem was addressed in Vieira Neto and Nehmzow’s work [13]. These authors use visual colored data, where visual attention based on salience maps was applied.

Considering change detection in 3D point clouds, some recent works were proposed. A combination of GMM and the Earth Mover’s Distance (EMD) [14] algorithm was proposed by Nez et al. [15] and extended in Drews et al. [3]. Despite the robustness of the obtained results, the computational cost is an important limitation of the method. In Nez et al. [9],
the structural matching algorithm was used instead of the EMD to determine changes in the space of the mixture of Gaussians. These methods are compared in [5] and the results show a small improvement in terms of computational load and the sensitivity to the number of Gaussians, but increasing the number of parameters of the change detection system. A fast version of the proposed algorithm in [3] is showed in [1] with similar accuracy.

Vieira et al. [4] introduced a method based on implicit functions to cluster the point clouds. It allows an efficient change detection algorithm using Boolean operations. The method uses a limited 3D occupation grids and a large number of parameters of the change detection algorithm. The sensitivity to the number of Gaussians, but increasing the number of Gaussians in the mixture decreases when the surfaces are not closed.

Andreasson et al. [2] presented a system for autonomous change detection with a security patrol robot using 3D laser range and a color camera, using the 3D-NDT structure [12] to model the data. The present work compares the proposed algorithm with the work of Andreasson et al. [2] in the context of geometric data to define which one is the best way to model point clouds in a probabilistic way using Gaussian distribution.

III. LEARNING FINITE GAUSSIAN MIXTURE MODELS

A. Gaussian Mixture Models

Let $X$ be a set of points of dimension $D$ and $k$-components of finite mixture distribution. The GMM probability function can be represented as the following:

$$
\Phi(x|\Theta_k) = \sum_{i=1}^{k} \pi_i \phi(x|\theta_i) \quad \forall x \in \mathbb{R}^D,
$$

where $\theta_i$ and $\pi_i$ correspond to the $i$-th component of the mixture and $\sum_{i=1}^{k} \pi_i = 1$. The vector $\Theta_k = \{\pi_1, \ldots, \pi_k, \theta_1, \ldots, \theta_k\}$ is the parameter set of mixture models, where $k$ is the number of components in the mixture.

The function $\phi = (x|\theta_i)$ represents the Gaussian probabilistic density function, given by

$$
\phi(x|\theta_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2} (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)},
$$

where $\mu_i$ is the $D \times 1$ mean vector, $\Sigma_i$ is the $D \times D$ positive definite covariance matrix and $\theta_i = (\mu_i, \Sigma_i)$. Given a point cloud $S = \{x_i\}_{i=1}^N$ with size $N$ and the mixture parameters $\Theta_k$, the maximum likelihood log can be calculated by follow equation:

$$
L(S|\Theta_k) = \log \prod_{t=1}^{N} \Phi(x_t|\Theta_k)
= \sum_{t=1}^{N} \log \sum_{i=1}^{k} \pi_i \phi(x_t|\theta_i).
$$

1In this work, $D$ value is equal to 3.

B. Expectation-Maximization Algorithm

The Expectation-Maximization (EM) algorithm [10] is the most popular technique that determines the parameters of a mixture from a set of points. Each iteration of the EM consists of two processes:

- **E-Step**: It computes the a posteriori probability of each sample in relation to each Gaussian of the mixture using the equation below.

$$
P(\theta_1, \pi_i|x) = \frac{\pi_i \phi(x|\theta_i)}{\sum_{l=1}^{k} \pi_l \phi(x|\theta_l)}.
$$

- **M-Step**: In this step, the components of the mixture are maximized. The updating of each $i$-th component is given by:

$$
\begin{align*}
\pi_i &= \frac{1}{N} \sum_{t=1}^{N} P(\theta_1, \pi_i|x_t), \\
\mu_i &= \frac{\sum_{t=1}^{N} P(\theta_1, \pi_i|x_t) x_t}{\sum_{t=1}^{N} P(\theta_1, \pi_i|x_t)}, \\
\Sigma_i &= \frac{\sum_{t=1}^{N} P(\theta_1, \pi_i|x_t) (x_t - \mu_k)(x_t - \mu_k)^T}{\sum_{t=1}^{N} P(\theta_1, \pi_i|x_t)}.
\end{align*}
$$

IV. THE PROPOSED METHOD

The proposed method is based on a modification of the EM algorithm called Split-EM [1] together with Spatial Difference Probability [2], where the a priori knowledge about the number of Gaussians $k$ in the mixture is needless and only the reference point cloud is modeled. Figure 1 shows an overview of the proposed approach.

Initially, in a pre-processing step, the point cloud is simplified using the PCL library [16]. Then, the Split-EM algorithm [1] is proposed to avoid local minimum and to estimate the number of Gaussians in the mixture. Finally, the changes in the GMM are found using Spatial Difference Probability. We detail each step below.
A. 3D Point Cloud Pre-processing

A crucial step of the proposed method is the pre-processing. It is responsible for reducing the size of the point clouds, removing redundant points. Point clouds are unordered data structures. So this problem is hard to solve because their topology is not available.

It’s important to consider that a point cloud with a large number of points influences in the algorithms execution time, even though the high computational complexity of the method. Two major problems related to point clouds are the outliers and the noises, acquired by sensors. They both affect the GMM estimation results.

The present work uses a library for point clouds manipulation, which is called Point Clouds Library (PCL) [16]. It contains numerous algorithms to manipulate this kind of data.

A simplification algorithm was used to reduce the number of points in the point clouds. It is based on 3D voxel grid [16]. After creating the voxels, the center of each voxel is computed. It represents the points inside the voxel. The resolution of the new point clouds depends on the voxel’s size. If it’s big, the point clouds are reduced, although some important information can be lost. In this work were used voxel’s size equals to 0.02 m3.

B. Selection Criteria

Selection criteria are used in the literate to identify an accurate model of a dataset. This model should optimize the performance and accuracy in relation to the others selection criteria.

It’s necessary to define criteria to distinguish between good and bad representations. For this, the entropy criterion is used to solve this problem.

Given a dataset $S$, the cost function MDL is defined as [11]:

$$C(\Theta_k|S) = \log \pi_k - \frac{n_2}{2} \sum_{i=1}^k \log \left( \frac{N \pi_i}{12} \right)$$

$$-\frac{k}{2} \log \left( \frac{N}{12} \right) - \frac{k(n+1)}{2}$$

where $n$ is the number of parameters of the model. Moreover, the correct parameters $\Theta_k$ correspond to the maximum of the function $C(\Theta_k|S)$, that is:

$$\Theta_k = \arg \max_{\Theta_k} \left( C(\Theta_k|S) \right)$$

C. Split-EM Algorithm

The Split-EM algorithm [1] estimates a Gaussian mixture model with important characteristics. It starts with one Gaussian in the model, set $\pi = 1$, $\mu$ and $\Sigma$ equals to the sample average and covariance, respectively. The algorithm splits the Gaussian that misrepresent its points. To do this, it needs to distinguish between good and bad representations. For this, the entropy criterion is used to solve this problem.

Entropy is a metric of disorder in a system. It is applied in fields such as physics, information theory and mathematics [18]. In general, the entropy of a random variable $X$ is given by:

$$H(X) = -\sum_{i=1}^N p(X = x_i) \log(p(X = x_i)).$$

Another important metric is the ideal entropy. It is an important limit for the entropy value. The ideal entropy of a Gaussian is given by:

$$H_{ideal} = \frac{1}{2} \log\left(2\pi e |\Sigma| \right)$$

Given a dataset $S = x_i^{N_{i=1}}$ and the probabilistic density function $\phi(x|\theta_i)$ of the $i$-th Gaussian of the mixture, it is possible to find the real entropy and the ideal entropy of this component. The equations to estimate them are:

$$H^i = \frac{1}{N_i} \sum_{t=1}^{N_i} \log(\phi(x|\theta_i))$$

$$H_{ideal}^i = \frac{1}{2} \log\left(2\pi e |\Sigma_1| \right)$$

The $S_{pi}$ is the difference between $H_{ideal}^i(X)$ and $H^i(X)$. It can be used to qualify how good the $i$-th component represents its points. The Gaussian in the mixture with the highest rate is splitted into two other Gaussians. The decision about where to split the component is made using the mean value of $\mu_i$ and the direction of the eigenvector $V_i$ with the highest eigenvalue $\lambda_i$ of the component. So, the new parameters are $\pi_{i1} = \pi_{i2} = \frac{\pi_i}{2}$, means equals to $\mu_{i1} = \mu_i + \sqrt{\lambda_i} * V_i$ and $\mu_{i2} = \mu_i - \sqrt{\lambda_i} * V_i$, and the covariance matrices equals to $\Sigma_{i1} = \Sigma_{i2} = \frac{1}{2} \Sigma_i$.

The algorithm has the following structure:

1) Initially, the GMM is composed by only one Gaussian. After that, the distribution is updated using EM algorithm and $C(\Theta_k|S)$ is computed.

2) Split: Given $k$ parameters $\Theta_k$, we split the Gaussian with lowest $S_{pi}$. The EM updates the parameters of $\Theta_{split}$, where the new GMM has $k+1$ Gaussians. Moreover, $C(\Theta_{split}|S)$ is computed.

3) If $C(\Theta_{split}|S) < C(\Theta_k|S)$ then the $\Theta_k$ is accepted and the algorithm is finished, else we back to the step 2.

D. Spatial Difference Probability

A probabilistic value of the point $x$ being different from the reference model is computed using the Gaussian reference model representation [2]. First, the Gaussian $i$ that contains the sample $x$ is found, i.e. using a threshold probability. This work set the threshold equals to 0.9. The Spatial Difference Probability is calculated using the Gaussian’s mean and covariance matrix as below:

$$p_{diff}(x) \approx e^{-(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

For each point in the point cloud, the $p_{diff}(x)$ is computed and a threshold is used to define if the point is consider change or not. The threshold is equal to 0.9.
V. CHANGE DETECTION WITH 3D-NDT

A. 3D-NDT

The basic principle of 3D-NDT is to represent the environment using a set of Gaussian distributions. It was introduced by Biber et al. [19] and first used to register 2D laser scanner. The method has been extended to 3D scan registration by Magnusson et al. [2] and recently a multi-resolution variant was presented by Magnusson et al. [12]. First, the considered space is divided into cells. Each cell that contains a certain minimum number of points is represented by a Gaussian with mean and covariance matrix computed from the points in the respective cell. The minimum number was set to 6 in the experiments presented by this paper. More formally, 3D-NDT is represented by a Gaussian with minimum number of points is represented by a Gaussian with mean and covariance matrix computed from the points in the respective cell. The minimum number was set to 6 in the experiments presented by this paper. More formally, 3D-NDT can be described as follows. We consider a point cloud $P = p_1, p_2, ..., p_n$ with points $p = [x, y, z]$ given in 3D Cartesian coordinates. The environment is divided into a set of cells $C = c_1, c_2, ..., c_k$ and for each cell $c_i$ a reference is stored to all $N_{ci}$ points $p_{ci}$ which lie within the cells boundaries. Then, for each cell $c_i$, the mean $\mu_{ci}$ and covariance $\Sigma_{ci}$ are calculated:

$$\mu_{ci} = \frac{1}{N_{ci}} \sum_{t=1}^{N_{ci}} p_t,$$

$$\Sigma_{ci} = \frac{1}{N_{ci} - 1} \sum_{t=1}^{N_{ci}} (p_t - \mu_{ci})^2.$$

Note that the 3D-NDT structure representation does not require evenly spaced data and hence can be calculated without further subsampling. All scans were subsampled with a 3D grid resolution of 0.1 m$^3$.

VI. EXPERIMENTS

In this section, the proposed change detection method has been analyzed in terms of robustness and computational load. The algorithms have been developed in C++ software and the benchmark tests are performed on a PC with processor Intel Core i7 3.4GHz with 12 Gb of DDR3 RAM and GNU-Linux Ubuntu 12.04. OpenCV and PCL libraries are used in the C++ implementation to improve the performance. The experiments were achieved in real environments inside an office area, where different sensors are used to acquire the point clouds. First, a Pioneer P3-AT with two SICK LMS-200 mounted orthogonally and running a DP-SLAM [20] was used in the UFMG dataset [3] (three different novelties: a cylinder, a person and a box). Second, a differential robot Robex equipped with a Hokuyo URG-30LX laser range finder in a pan-tilt unit has been used for acquiring the UNEX dataset [21] (two different changes in the scene: a person and a box). Finally, a PatrolBot robot equipped with a Microsoft’s Kinect motion-sensing system used in the FURG dataset, where the change is a person sitting in a chair inside a cluttered office environment. Finally, to obtain statistical significant results, the experiments were repeated five times for each test area. Further details about the UFMG and UNEX datasets can be obtained in [5].

In the Fig. 2, we show a 3D point cloud from FURG dataset. Fig 2(a) shows the reference 3D point cloud as an illustrates textured one. Fig. 2(b) show a current 3D point cloud. The photometric information is not used in the present work.

A. Quantitative Assessment

In order to evaluate the proposed method, the ground truth has been generated as follows: first, the different geometrical shapes have been manually selected. This manual segmentation is done using a C++ software implemented by the authors, where the topology and geometry information of the point cloud is considered in the process. Fig. 2(c) illustrated this process. In order to obtain a statistical quantitative assessment of the change detection methods, the methodology proposed by Vieira Neto and Nehmzow [22] is used. They propose the use of the index of agreement $\kappa$ that quantifies the strength of the data association process, where the values close to zero represent weak associations and the values near to one indicate strong associations. Therefore, the index $\kappa$ varies between $[-1; 1]$, where values smaller than 0.1 represent no agreement; values between 0.1 and 0.4 correspond to weak agreements; values between 0.4 and 0.6 represent a clear agreement and values larger than 0.6 represent strong agreements.

B. Evaluation of change detection method

In this section, an evaluation of the proposed method is compared with the approach proposed by Andreasson et al. [2]. These methods are divided into three steps, the first step is related to build a model and the second is related to compare them. The third step is the same for both methods, although the proposed method is faster due the low number of Gaussians in the model. We evaluated the methods in a qualitative and quantitative way.

Firstly, the qualitative results are obtained using the FURG dataset, see Fig. 2. Initially, we modeled the 3D point cloud using the Andreasson’s approach, where Fig. 3(a) shows the model, where the Gaussians are represented by red ellipsoids. With the view to evaluate the sensibility of the voxel’s size, a critical parameter in the method, we change it from 0.1 m$^3$ to 0.01 m$^3$, in figs. 3(b) and 3(c). Considering the size equals to 0.1 m$^3$, the change is detected with some outliers. If the size is defined as 0.01 m$^3$, almost all points are considered as change. The method’s problem is to define the voxel’s size. Its not possible to detect change with high voxels size. If it’s a small voxel, the algorithm is adept to detect change but with some outliers.

Fig. 4 shows the results obtained using the proposed algorithm. The Split-EM segments the point cloud automatically, as illustrated by Fig 4(a), where the Gaussians are represented by different colored ellipsoids. Fig. 4(b) shows the results of the detection step, where the change is correctly detected. It is shown by the white points in the figure.

The quantitative assessment was achieved using the UFMG and the UNEX dataset. The performance of the proposed system is shown in Table I and Table II related to the UFMG and UNEX datasets, respectively. The number of points in the point cloud of both datasets are similar. However, the performance is quite different. They differ mainly due to the scene. In the UFMG dataset, the environment is simpler than the UNEX, where the environment is composed by walls and the roof. Although, the roof in this case is composed by a
Fig. 2. Point Clouds acquired with FURG dataset: a) Reference 3D point cloud, b) Current 3D point cloud where a person appears in the scene and c) Ground truth manually generated, where the yellow points meaning the change. It is important to call attention that we use a geometrical information from the point clouds.

Fig. 3. Change Detection using the 3D-NDT method: a) Model created using voxel’s size equals to 0.1 m³, where the Gaussians are represented by red ellipsoids; b) Result of the change detection with the voxel’s size equals to 0.1 m³; c) Result of the change detection with the voxel’s size equals to 0.01 m³. In these cases white points represent changes in the environment and black points represent the unchanged points.

Fig. 4. Change Detection using the proposed method: a) Model created using the Split-EM algorithm, where the Gaussians are represented by different colored ellipsoids; b) Results of the change detection using the proposed method, where white points represent the detected changes.

set of pipes with parallax problem. And, the UNEX dataset was acquired in an office environment with a set of different objects.

The 3D-NDT algorithm is very sensible to the voxel’s size, as shown in Fig. 3. On the other hand, the Split-EM algorithm is not sensible because the method determines the number and parameters of the Gaussians according to the geometric information of the data. The results in the Tables I and II show that the proposed method is faster than the Andreasson’s approach. The time elapsed of the proposed method to build the model is high but the detection time is small. It is valid for all datasets.

In the UFMG datasets the proposed method is able to detect the change, as illustrated by the Kappa index ($\kappa$). The 3D-NDT algorithm is capable to detect in a better way for two different changes, i.e. with a small number of outliers. In the case of the person, our method detects better. For the UNEX datasets, our method is able to detect changes in a much better way and the elapsed time is five times smaller, in the case of the box.

<table>
<thead>
<tr>
<th>TABLE I. Evaluation of the change detection method proposed in the UFMG dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of Map</td>
</tr>
<tr>
<td>UFMG - Cylinder</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UFMG - Person</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UFMG - Box</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. Evaluation of the change detection method proposed in the UNEX dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of Map</td>
</tr>
<tr>
<td>UNEX - Box</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UNEX - Person</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The 3D-NDT algorithm is very sensible to the voxel’s size, as shown in Fig. 3. On the other hand, the Split-EM algorithm is not sensible because the method determines the number and parameters of the Gaussians according to the geometric information of the data. The results in the Tables I and II show that the proposed method is faster than the Andreasson’s approach. The time elapsed of the proposed method to build the model is high but the detection time is small. It is valid for all datasets.
This article describes a new method to detect changes in 3D point clouds. Gaussian mixture models are efficiently obtained using an iterative combination of a split paradigm and a cost function. Furthermore, we model only the reference 3D point cloud and compare it with the current model using the Spatial Difference Probability. It makes our method very fast, if compared with the previous approaches. Experimental results demonstrate that the best method is the Split-EM algorithm, both in terms of computational burden and robustness.

Moreover, the proposed method may be easily extended to detect things removed from the scene, which is achieved by simply using the current map as the reference input. The technique presented in this work opens new opportunities in automatic surveillance of infrastructures, by proposing a robust approach for searching and detecting changes. The applicability in mobile robots was evaluated showing the capabilities of the proposed method to be applied in real robots located by odometry and laser-based SLAM. The robot localization is an important constraint, but it has smaller impact in the performance than point-to-point approaches.

Future works will be focused on the use of the current novelty detection algorithm in fields like surveillance or exploration of dangerous environments. Also, an extension of the Gaussian mixture models estimation method will be developed to work iteratively, so that data can be captured and processed online by the mobile robot.

ACKNOWLEDGMENT

The authors would like to thank RoboLAB and NAUTEC members for their collaboration in this work, especially, RoboLAB member Dr. Pedro Núñez from University of Extremadura and NAUTEC member Pedro Ballester from Federal University of Rio Grande. Finally, thanks to the support of CNPq, FAPERGS, FURG and FINEP.

REFERENCES


