

Spatial Density Patterns as Representation for Point Cloud Comparison and Applications

Antonio Wilson Vieira and Mario Fernando M. Campos

Departamento de Ciéncia da Computaçao

Universidade Federal de Minas Gerais

Belo Horizonte, Brazil

Email: {awilson,mario}@dcc.ufmg.br

Abstract—In this work, spatial density maps are proposed as a high-level representation for comparing objects given by point clouds. Traditional approaches present complex metrics, which includes statistical information associated with the underlying point distribution. Our method obtains a representation by spatial density maps in linear time, allowing query in constant time and efficient comparison between different sets.¹

Keywords-point cloud, change detection, pattern recognition;

I. INTRODUCTION

For a large number of applications with 3D point clouds, a high level representation for the points need to be built. For instance, comparison between different sets are usually employed using some clustering of points in basic primitives to allow comparison operations on the higher level. In this work, we address some problems of point cloud comparison, where the choice of the representation and similarity metrics determine the efficiency of the methods and the robustness of the results.

As contribution of our work, we present a representation that allow for efficient comparison and Boolean operations among point clouds based on spatial density. We show that our representation using spatial density patterns, when compared with literature, has the following advantages:

- Linear time construction;
- Level-of-detail storage;
- Constant time query;
- Linear time comparison;
- Basic primitives independence.

This work also describes applications of our representation using spatial density patterns to problems in the literature where a representation by Gaussian mixture models is replaced by spatial density patterns representation with gains in efficiency and robustness. The first application is the change detection in point clouds, which is also a topic of interest in mobile robotics and monitoring. The second application deals with the recognition of human actions in sequences of point clouds which is a topic that has gained importance with the popularization of real-time 3D sensors as the *Kinect*.

II. RELATED WORK

Most applications with point clouds requires a higher level representation for comparing different sets. The representation of implicit surfaces, mixture of Gaussian and segmentation into basic primitives are widely used as high-level representation for comparison of point clouds.

The representation of implicit surfaces from point cloud produces a triangular mesh when the data is well organized in order to define a surface, usually with known normals for all points. Several recent works on surface reconstruction construct a scalar field where zero level defines an implicit surface to fit the points cloud [1], [2].

The segmentation into basic primitive provides a high-level representation for point clouds, extracting geometric primitives such as planes, cylinders, ellipsoids, etc.. Works like [3] proposes the extraction of primitives as planes, cones and cylinders in point clouds. In [4] is proposed a segmentation of a set of ellipsoids.

Several studies, such as those reported in [5], [4], [6], propose the use of Gaussian mixture models (GMM) to model the point cloud in order to compare and detect changes. However, the technique becomes computationally expensive due to the use of the EM algorithm in the construction of GMM because the parameters need to be estimated for each new point cloud. In addition, the shapes are limited to a few basic primitives like quadrics and superquadric [7]. In [6], recognition of human actions from point clouds is modeled by the use of GMM as representation, where the Hausdorff distance is used to compute the dissimilarity between two clouds of points. In our work, a representation based on spatial density patterns replaces GMM for both problems of change detection and human action recognition from point cloud.

III. METHODOLOGY

We will now describe in further detail how to build our global density function from a point cloud set and how to use the implicit function theorem to define a surface that clusters points according to spatial density values. Finally, Boolean operations over two point clouds are presented using their corresponding density functions.

¹This work relates to a Ph.D. thesis

A. Density Function from Point Cloud

The main idea underlying our method is the estimation of an unknown global density function $G : \mathbb{R}^3 \rightarrow \mathbb{R}$ from a given point cloud $P = \{(x_i, y_i, z_i), 1 \leq i \leq n\}$ such that larger values occur at points closer to the surface defined by P . This global density function defines a scalar field over which an isovalue density threshold h can be used to cluster points in the implicit volume defined by the set $S = \{p \in \mathbb{R}^3; G(p) \geq h\}$.

Given a point cloud set P with n points, we define, for each point $p_i \in P$, a simple local density $g_i : \mathbb{R}^3 \rightarrow \mathbb{R}$ which is a kernel function chosen to be a Radial Basis Function (RBF) to weight the contribution of p_i to the global density.

In this work we adopt a Gaussian RBF kernel and the parameter σ is adjusted to smooth the influence of p_i over its neighborhood. We set σ with the value of the maximum Euclidean distance between two given points $p_i, p_j \in P$ so they are considered neighbors. Hence, if p_i and p_j are neighbors, which means $\|p_i - p_j\| \leq \sigma$, then the density contribution δ between p_i and p_j is given by

$$\delta = g_i(p_j) = e^{-\frac{\|p_i - p_j\|^2}{2\sigma^2}} \geq e^{-\frac{1}{2}}. \quad (1)$$

In order to define the global density function G , we accumulate the local density functions g_i contributed by all points p_i , $1 \leq i \leq n$,

$$G(p) = \sum_{i=1}^n g_i(p). \quad (2)$$

Since the local density function $g_i(p_i) = 1$ for all $p_i \in P$, if p_i has k neighbors, then $G(p_i) \geq 1 + k\delta$. Our method clusters all points with at least k neighbors by setting a threshold value $h = 1 + k\delta$. Fig. 1 illustrates our clustering strategy for a 1D point cloud. The thick blue line delimits the clustered set of points with at least two neighbors, which include p_2 and p_3 . Notice that, given the threshold value $h = 1 + 2\delta$, p_1 , p_4 and p_5 have no neighbors and does not belong to the cluster.

Increasing the smoothing parameter σ leads to increasing neighborhood radius causing more points to be clustered together. Fig. 2 illustrates this cluster growing effect for a 2D point cloud.

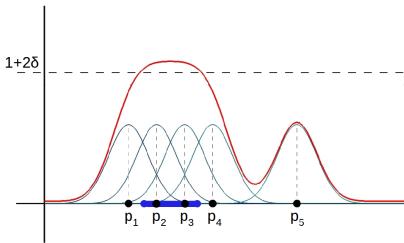


Fig. 1. Example of point clustering in 1D. The local density functions (thin blue) are accumulated to define a global density function (red). thick blue line resumes the 1D cluster set of points whose accumulated density is larger than $1 + 2\delta$. Notice that, given the threshold, the accumulated density for p_1, p_4 and p_5 is insufficient to cluster them.

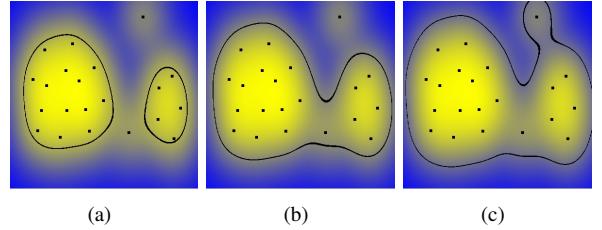


Fig. 2. Example of the point clustering in 2D. Increasing values of the smoothing parameter σ leads to an increased cluster volume, causing more points to be clustered together. In (a), two points are considered outliers, in (b) only one point is considered as outlier and, in (c), no outliers are considered.

In order to speed up computation, we obtain a discrete sampling for the global density function by computing its values for a 3D grid defined for the bounding box of the point cloud P . For each point $p_i \in P$, its local density contribution is calculated and summed only for a constant number of grid points on its neighborhood, according to the kernel mask size s . So, the global density function G is sampled, for all points of the grid, in time $O(n)$ for a point cloud with n points.

The grid values is then used as a discrete sampling for G . In order to obtain a continuous approximation for G in all points from the bounding box of the point cloud P , we use a method of multivariate interpolation on a 3-dimensional regular grid [8]. Using such a strategy, the global density function G is constructed in linear time and a particular value $G(p)$ can be estimated in constant time for each point p in the bounding box. Furthermore, storage complexity doesn't depend on the number of points on the cloud, but on the grid size. The grid size is related to the desired level of detail for density sampling that defines the numbers n_x, n_y and n_z of segments that the bounding box is divided along x, y and z axes, respectively. We denote by $n_x \times n_y \times n_z$ the grid size and by l the length of each grid segment.

B. Clustering by Implicit Surface

From the theorem of implicit functions, we have that if $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ is smooth and $h \in \mathbb{R}$ is a regular value, that is, $\nabla f(p) \neq 0$ for all $p \in f^{-1}(h)$, then $f^{-1}(h)$ is a surface that partitions the domain \mathbb{R}^3 into three sets:

$$S_1 = \{p \in \mathbb{R}^3; f(p) < h\}, \quad (3)$$

$$S_2 = \{p \in \mathbb{R}^3; f(p) = h\}, \quad (4)$$

$$S_3 = \{p \in \mathbb{R}^3; f(p) > h\}. \quad (5)$$

The global density function G constructed as described in the previous section is a smooth scalar function in \mathbb{R}^3 , and by setting a constant threshold $h > 1$, we define the set $S_2 \subset \mathbb{R}^3$ as the boundary of a continuous volume region that clusters the point set P . Fixing $h = 1 + k\delta$, we define a set to cluster points that have at least k neighbors whose distance is less or equal than σ . This set can be implicitly defined as $S = \{p \in \mathbb{R}^3; G(p) > h\}$.

If h is not a regular value for G , we do not have a manifold surface due to possible singularities, however the cluster is

still well defined. In order to represent and visualize the set S_2 as a manifold surface, we use a special version of the Marching Cubes algorithm [9] to extract a piecewise linear boundary surface with topological guarantees. The Marching Cubes method produces a mesh of triangles approximating the inverse image $G^{-1}(h)$ from a grid of cubes for which the values of the function G at their vertexes are known. Each cube vertex on the grid is classified either as positive or as negative, according to its relative position with respect to the set S . The resulting mesh represents a boundary surface that clusters the raw point cloud. Different level sets can be extracted for the same point cloud as long as we use a different isovalue h .

Figure 3-a shows an example of a point cloud with some outliers; its level set surface extracted for a global density value $h = 1 + 2\delta$ is depicted in Figure 3-b, and another level set surface extracted for density level $h = 1 + 3\delta$ is shown in Figure 3-c. Notice that multiple components are well clustered. Notice, also, that in Figure 3-c the larger density level clustering leads to a disconnected component for the person's head and that in the lower members, a hole appeared separating the two legs.

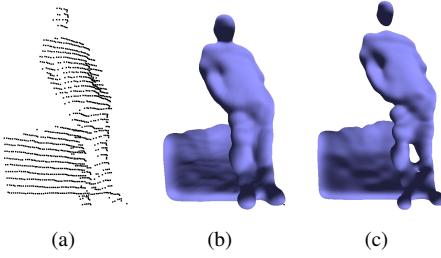


Fig. 3. Example of the point cloud clustering in 3D. Raw point cloud (a), the clustering set for density level $h = 1 + 2\delta$, and (b) the clustering set for density level $h = 1 + 3\delta$ (c).

In our experiments, the implicit volume is defined so as to cluster points with at least $k = 3$ neighbors. Therefore, we use the threshold $h = 1 + 3\delta$ where $\delta = e^{-\frac{1}{2}}$. The smoothing parameter will be set as $\sigma = 2l$ where l is the voxel length defined according to the grid size.

Given a threshold h , we will consider a normalized global density function $\mathcal{G}(p) = h - G(p)$, where zero value defines the boundary surface, negative and positive values define interior and exterior regions of the cluster, respectively. The use of \mathcal{G} allows, without loss of generality, for the efficient use of Boolean operations on implicit volumes, which we will describe next.

C. Boolean Operations

Let two point clouds P_1 and P_2 represent current and reference data, respectively. It is hard to perform Boolean operations such as $P_1 \cap P_2$ or $P_1 - P_2$. However, using their associated normalized global density functions, given by $\mathcal{G}_1, \mathcal{G}_2 : \mathbb{R}^3 \rightarrow \mathbb{R}$, each Boolean operation constructs a new normalized global density function \mathcal{G}_3 in a simple step, as shown in [10], using the following operations:

$$P_1 \cup P_2 \Rightarrow \mathcal{G}_3(p) = \min(\mathcal{G}_1(p), \mathcal{G}_2(p)), \quad (6)$$

$$P_1 \cap P_2 \Rightarrow \mathcal{G}_3(p) = \max(\mathcal{G}_1(p), \mathcal{G}_2(p)), \quad (7)$$

$$P_1 - P_2 \Rightarrow \mathcal{G}_3(p) = \max(\mathcal{G}_1(p), -\mathcal{G}_2(p)). \quad (8)$$

Fig. 4 illustrates the *difference* Boolean operation between two scalar fields. In (a) and (b), the sets P_1 e P_2 are defined by the scalar fields \mathcal{G}_1 and \mathcal{G}_2 . In (c) the maximum among \mathcal{G}_1 and $-\mathcal{G}_2$ results in negative values for points related to the difference, leading to the segmentation of the change.

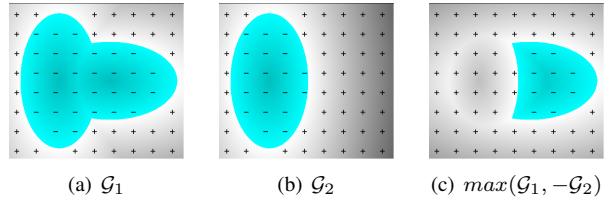


Fig. 4. Example of the *difference* Boolean operation between two scalar fields. Negative values define the interior of each set and the difference Boolean operation is obtained by simple combination of the sign of the scalar fields.

The above Boolean operation is defined for implicit data only. In order to take advantage of this procedure to compute changes from our discrete point clouds, we obtain $\mathcal{G}_3(p) = \max(\mathcal{G}_1(p), -\mathcal{G}_2(p))$, which is a new density function whose negative values define a volume where points belonging to the portion where there is a change should lie. Finally, points $p \in P_1$ are detected as a change if $\mathcal{G}_3(p) < 0$, which is a constant time query, leading to linear time computation for the set P_1 .

Algorithm 1 details our method. Notice that, as stated, Algorithm 1 computes change points that appears between acquisitions. Disappearing change points can be computed as well by swapping P_1 and P_2 in the algorithm's input.

Algorithm 1 ChangeDetect(P_1, P_2, h)

- 1: Compute \mathcal{G}_1 from P_1
 - 2: Compute \mathcal{G}_2 from P_2
 - 3: Compute $\mathcal{G}_3 = \max(\mathcal{G}_1, -\mathcal{G}_2)$
 - 4: $D = \emptyset$
 - 5: **for** each $p \in P_1$ **do**
 - 6: **if** $\mathcal{G}_3(p) < 0$ **then**
 - 7: $D = D \cup \{p\}$
 - 8: **end if**
 - 9: **end for**
 - 10: **return** D
-

IV. EXPERIMENTAL RESULTS

In order to obtain a statistical quantitative assessment of the proposal method, we used the methodology proposed by Vieira Neto and Nehmzow [11] to evaluate the κ index of change detection. The κ index varies between $[-1; 1]$, where values smaller than 0.1 indicate no agreement; values between

0.1 and 0.4 indicate a weak agreement; values between 0.4 and 0.6 indicate a clear agreement, and values larger than 0.6 indicate a strong agreement. More details on these metrics can be found in [11].

A. Evaluation of robustness and parameters sensitivity

The experiments to evaluate robustness of our method used a 3D dataset acquired in an office environment as the reference map and 10 different datasets of the same environment modified with objects of different sizes and shapes. A Pioneer 2-AT Robot equipped with a Microsoft Kinect sensor was used in data acquisition, and each data set has about 76,000 points. The environment and the 10 different changes are shown in Figure 5-a and 5-b. Figure 5-c shows a reference 3D map of the environment and Figure 5-d shows a 3D map for change 1 inserted in the environment. Tests were executed on a PC with a Core 2 Duo CPU running at 2.0 GHz and with 2 GB RAM.

Our algorithm obtains a discrete sampling for the global density function by computing its values for a 3D grid defined for the bounding box of the point clouds. Larger grid sizes leads to dense sampling, which improves robustness but decreases time performance, as shown in Figure 6.

Firstly, Figure 6-a depicts the statistical results (mean and standard deviation (error bar)) obtained with the proposed algorithm using the κ index. These results were generated on a grid $40 \times 40 \times 40$ to $320 \times 320 \times 320$, with a step of 40. The result shows that high accuracy is obtained from grid size $80 \times 80 \times 80$ and that this accuracy is stable for higher values. Provided that values greater than 0.6 indicate strong agreement, we are able to detect changes even for small sized changes (see Figure 5-b).

Figure 6-b, shows the processing time of the proposed algorithm. The processing time is obtained considering the total time for each of the three steps: (1) building implicit volume for reference point cloud, (2) building implicit volume for the point cloud with changes and, (3) performing the Boolean operation (change detection). The time results are shown with mean and standard deviation. This graph shows that increasing grid size, increases computational time without any gain in accuracy for grid sizes larger than $80 \times 80 \times 80$, for which high accuracy is obtained in less than half a second.

In order to experimentally show the linear time complexity of our method, we computed changes for point clouds of

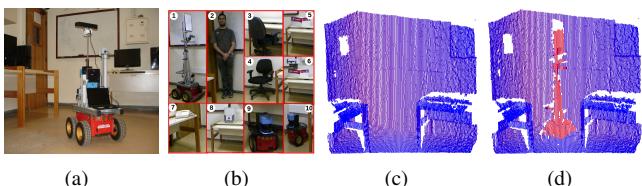


Fig. 5. Experimental Setup 1: (a) Environment and experimental setup composed of a Pioneer Robot equipped with a Kinect sensor used to acquire 3D data. (b) Ten different changes inserted in the office environment of various size, pose and shape. (c) Reference 3D map from the environment. (d) 3D map from change 1 inserted in the environment.

different sizes. For this experiment, nine sets were used for which the number of points varied from 17,000 to 144,000. For each dataset, five different acquisitions were processed to present the mean and standard deviation of the processing time as shown in Figure 7.

B. Comparison with a state-of-art approach

We use the same dataset to compare our method with the one presented in Drews et al. [7] that uses a combination of GMM and EMD to detect changes retrieved as superquadric shapes. While our method has the grid size as an important parameter that affects both performance and accuracy, their method is highly sensitive to the number of Gaussians that are used to represent the point cloud. Hence, we present their performance and accuracy results for different number of Gaussians. Figure 6-c and 6-d show the accuracy and the time performance for 8 different number of Gaussians using [7]. We notice that, although processing time increases with the number of Gaussians (Figure 6-d), the accuracy doesn't improve consistently while increasing this parameter (Figure 6-c), and it achieves the best result with 9 Gaussians distributions. Furthermore, while our method achieves high accuracy in less than half a second, theirs spend more than 10 seconds to achieve their best accuracy with 9 Gaussians.

We also compared our method with the one proposed by Nuñez et al. [12], that extends [7], by using a structural matching algorithm, instead of EMD, to determine changes in GMM space. For this experiment we used the same dataset as used by [12]. In their experiments, three different novelties were included in the robot's workspace: An identification cylinder, a person and a printer box. The environment and the objects used are shown in Figure 8-a and 8-b. That dataset was acquired in the hallways of a building of the *Universidade Federal de Minas Gerais* using a Pioneer 2-AT Robot equipped with two SICK LMS-200 mounted orthogonally, in order to localize and to acquire 3D maps.

For this experiment, we considered the time processing spent to attain a strong agreement in change detection using Nuñez et al. [12] and our method. As [12], we are also capable to detect changes in all three scenarios, but faster. Table I uses the best times obtained in the work of Nuñez et al. [12]. Both methods are divided into two steps, where the first step

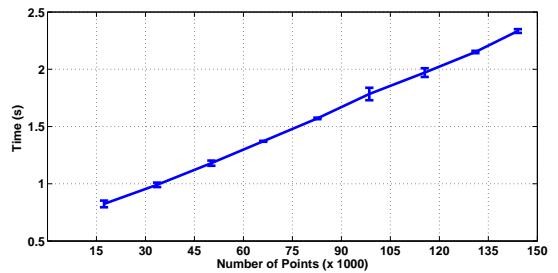


Fig. 7. Computation time for different point cloud sizes. Nine sets with different sizes from 17,000 to 144,000 were used to show the linear time complexity in the number of points.

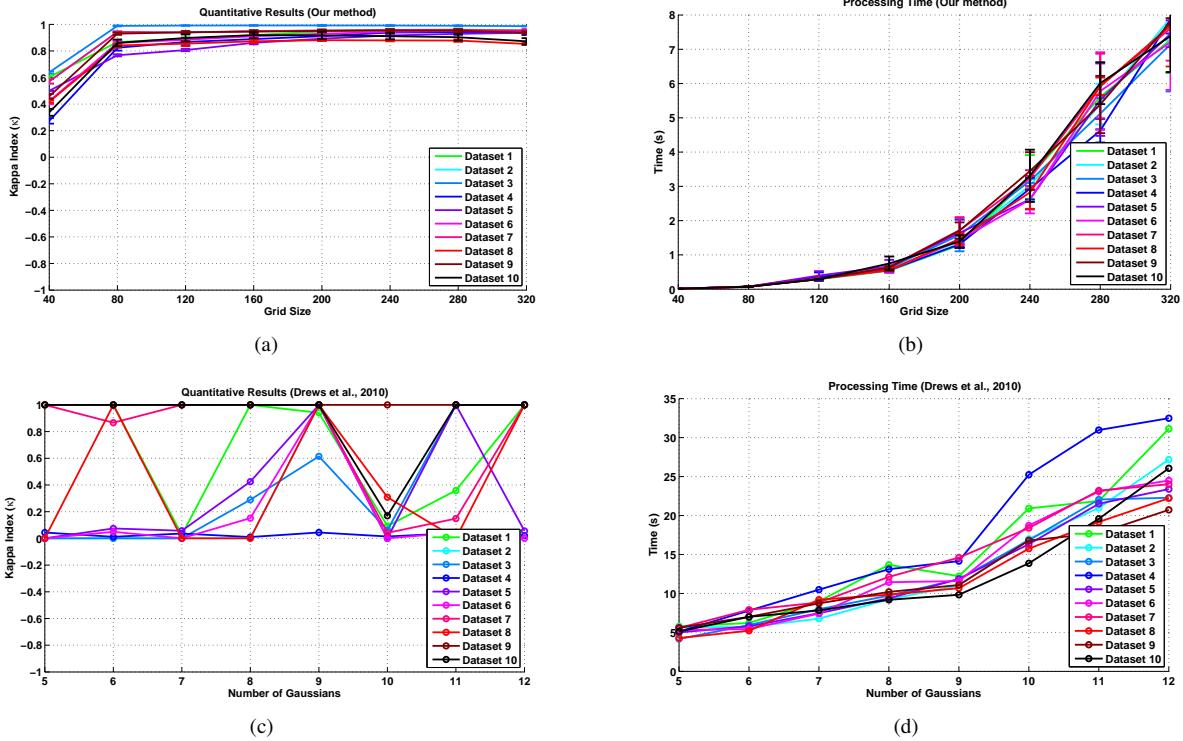


Fig. 6. Results obtained using different datasets acquired with the Kinect, with objects of different sizes and shapes and for different poses. (a) Index of agreement κ using our method and varying grid size parameter. (b) Processing time using our method and varying grid size parameter. (c) Index of agreement κ as used in [7], but varying the number of Gaussians. (d) Processing time as used in [7] and varying the number of Gaussians.

TABLE I
COMPARATIVE STUDY OF DIFFERENT CHANGE DETECTION ALGORITHMS.

| Dataset | Number of Points | | Build Ref. Map | | Build Cur. Map | | Detection | |
|-------------------------|------------------|----------|----------------|------|----------------|------|--------------|-------|
| | Ref. Map | Cur. Map | Nuñez et al. | Our | Nuñez et al. | Our | Nuñez et al. | Our |
| Real Data - Test Area 1 | 79171 | 79633 | 177.21 | 1.12 | 164.56 | 1.07 | 0.014 | 0.146 |
| Real Data - Test Area 2 | 79171 | 81134 | 177.36 | 1.16 | 110.86 | 1.05 | 0.014 | 0.149 |
| Real Data - Test Area 3 | 79171 | 80112 | 167.88 | 1.14 | 109.22 | 1.06 | 0.028 | 0.166 |

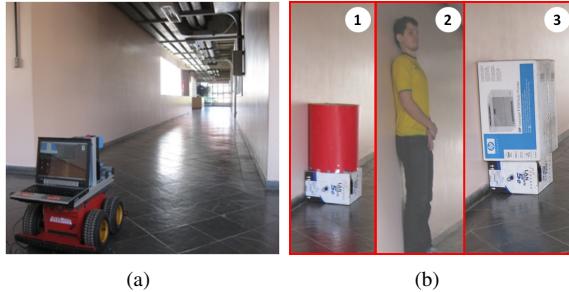


Fig. 8. Experimental Setup 2: (a) Environment and experimental setup composed of a Pioneer 2-AT Robot equipped with two orthogonally mounted laser scanners acquiring 3D data. (b) Three different changes were inserted in the robot's environment.

builds a “special” representation and the second step compares this representation (called Detection). In the case of [12], the representation is built using a simplification method and GMM estimation. In our method, the representation is tailored to building implicit volumes. The change detection is achieved using structural matching in [12], and Boolean operation in our approach.

Notice that for all datasets, our method is substantially faster to build the required representation. The main limitation of the method proposed by Nuñez *et al.* [12] is related to the Gaussian Mixture Models using the Expectation-Maximization algorithm, which has lower computational performance and is quite sensitive to the number of Gaussians used, which is one of the parameters of the EM algorithm.

Figure 9 gives an example of the dataset used in [12]. Figure 9-a shows a person that is the change in the environment, Figure 9-c shows the reference map without the change, Figure 9-d shows the actual map with segmented change. The blue dots represent the points detected as no change, and the red points as changed points detected by our algorithm. Some red points are in the ceiling, and this is due to the parallax effect in the acquisition and the misalignment between the reference dataset (without change) and the present dataset (with change).

Besides the improvement in computational time processing, we observe from the results as shown in the Figure 9, that our method performs quite robustly in the presence of parallax and highly complex geometry. In the work of [12], the datasets obtained by the robot show the walls and some objects, but

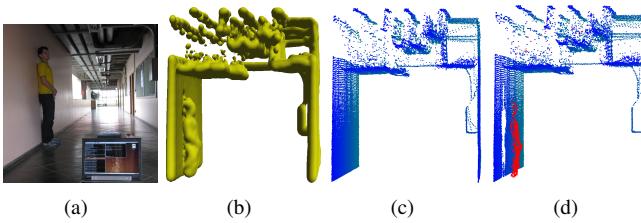


Fig. 9. Details of the experimental setup 2: (a) A person inserted as the change in the environment. (b) Implicit volume obtained for the dataset. (c) Reference 3D map of the environment. (d) Current 3D map of environment with inserted change.

the roof was also acquired, where pipes and other objects are present in the ceiling, as shown in Figure 8-a. However, this data are not considered in their experiment, because of the high data complexity due parallax and absence of meaningful changes. In another experiment, we used the “complete” information available from this dataset to describe the data including all the points. This point cloud is approximately twice as large as the partial volume, *i.e.* it has approximately 150,000 points. We were able to correctly detect changes for that dataset in less than one second. Figure 9-b shows the clustered volume obtained with our implicit approach for the complete dataset.

V. CONCLUSIONS AND FUTURE WORK

This work described a novel method to detect changes in a 3D real environment for autonomous surveillance robot by combining density calculation and 3D Gaussian smoothing to transform the raw point cloud data into a continuous 3D density field, upon which Boolean operations are used to express the concept of change detection between the two clouds. By using simple Boolean operations on the smoothed density space, our method achieves quite a complex operation in a simple way, leading to both robustness and computational speed.

Indeed, comparisons showed that in spite of its simplicity, our method outperforms state-of-art methods in accuracy, and it is efficient enough to run an order of magnitude faster than a comparable method from the literature. Furthermore, our method performs well for high complex datasets and it does not rely on any assumptions as to how the data was acquired, making it widely applicable for range images or unstructured 3D data point clouds.

Current implementation of our method constructs the density field using a 3D grid to obtain a discrete sampling for this field. Although continuity is recovered by using multivariate interpolation, such strategy depends on the storage of the grid values, whose complexity is cubic on the grid size. If memory storage is not an issue, this strategy will be efficient due to its linear time complexity to construct the density field representation and linear time comparison using Boolean operation. However, in order to avoid such discrete sampling and memory storage, one can construct the density field by learning a subset of points and the associated kernels with

optimized weights to best approximate the desired density field using least squares. Despite being more time consuming, this strategy leads to a mathematical model of the density field that avoids using grid storage.

Future investigation will focus on the extension of the method to work without grid sampling and to run iteratively, with the data being captured online by the robot. Another important improvement is to perform the registration between clouds concurrently with Boolean operation. This will ultimately allow us to correlate and align similar points and obtain good registration even for significant changes in the environment and in the acquisition conditions.

ACKNOWLEDGMENT

The authors would like to thank this colleague and this financing institute.

REFERENCES

- [1] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, “Multi-level partition of unity implicits,” in *Proceedings of conference on Computer graphics and interactive techniques (SIGGRAPH)*. ACM, 2003, pp. 463–470. [Online]. Available: <http://doi.acm.org/10.1145/1201775.882293>
- [2] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Symposium on Geometry processing (SGP)*. Eurographics, 2006, pp. 61–70.
- [3] P. Rodríguez González, D. González Aguilera, and G. Gómez Lahoz, “From point cloud to surface: Modeling structures in laser scanner point clouds,” in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS*, vol. 3, 2007, pp. 338–343.
- [4] F. Pauling, M. Bosse, and R. Zlot, “Automatic segmentation of 3d laser point clouds by ellipsoidal region growing,” in *Proc. of Australasian Conf. on Robotics and Automation, 2009*, 2009.
- [5] P. Núñez, P. Drews, R. Rocha, M. Campos, and J. Dias, “Novelty detection and 3d shape retrieval based on gaussian mixture models for autonomous surveillance robotics,” in *Proceedings of International Conference on Intelligent Robots and Systems*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4724–4730.
- [6] W. Li, Z. Zhang, and Z. Liu, “Action recognition based on a bag of 3d points,” in *CVPR Workshop for Human Communicative Behavior Analysis*, June 2010.
- [7] P. Drews Jr, P. Núñez, R. Rocha, M. Campos, and J. Dias, “Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots,” in *Proceedings of International Conference of Robotics and Automation*, may 2010, pp. 3635–3640.
- [8] Y. Bai and D. Wang, “On the comparison of trilinear, cubic spline, and fuzzy interpolation methods in the high-accuracy measurements,” *Trans. Fuz Sys.*, vol. 18, no. 5, pp. 1016–1022, Oct. 2010. [Online]. Available: <http://dx.doi.org/10.1109/TFUZZ.2010.2064170>
- [9] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, “Efficient implementation of marching cubes cases with topological guarantees,” *Journal of Graphics Tools*, vol. 8, no. 2, pp. 1–15, december 2003.
- [10] A. Ricci, “A constructive geometry for computer graphics,” *The Computer Jurnal*, vol. 16, no. 2, pp. 157–160, 1973.
- [11] H. Vieira Neto and U. Nehmzow, “Visual novelty detection for autonomous inspection robots,” in *Service Robot Applications*, Y. Takahashi, Ed. Vienna, Austria: I-Tech Education and Publishing, 2008, pp. 309–330.
- [12] P. Núñez, P. Drews, A. Bandera, R. Rocha, M. Campos, and J. Dias, “Change detection in 3d environments based on gaussian mixture model and robust structural matching for autonomous robotic applications,” in *Proceedings of International Conference on Intelligent Robots and Systems*, oct. 2010, pp. 2633 –2638.