POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Anomaly detection in point clouds with Local Polynomial Approximation & Intersection of Confidence Intervals

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Riccardo Nobili

Advisor: Prof. Giacomo Boracchi

Co-advisors: Diego Stucchi, Luca Magri

Academic year: 2021-2022

1. Introduction

Given the recent major improvements that have occurred in technologies that enable surface scanning, point clouds are starting to become an increasingly powerful tool. Thanks to the great precision of the scanners, they are able to catch all the smallest details and geometrical features and so they can offer a very simple and easily understandable representation of the scanned objects. Point clouds allow you to have detailed scans of objects and surfaces of every kind, from the smallest and common ones to big and complex infrastructures like buildings, bridges and roads. The detection of defects is fundamental in any field, especially in the civil one where routine checks on buildings and monitoring on their conditions are critical tasks, but necessary in order to prevent catastrophic failures. Thanks to the accuracy of such scans it is possible to precisely capture the underlying geometry of objects and to use point clouds to search for defects and to identify malformations on the analyzed surfaces. Presenting a new method such as the one we propose is therefore very important in this area that is still too little explored, but which is very delicate and often critical.

The importance of working in this field relies on the fact that techniques like the one described in this thesis are quite innovative: there are a lot of works dealing with the search of defects and malformations in 2D representations (images) of objects, but very few on their 3D representation as point cloud. This is probably due to the lack of comprehensive 3D datasets that are explicitly designed for the unsupervised detection and localization of anomalies directly on point clouds. Working directly on point clouds to search for defects also allows to speed up and make safer operations involved in the searching of defects on structures that usually requires people to work in dangerous situations, potentially exposed to hazardous conditions. For this reason, a typical approach is to reconstruct the 3D model of the surface that is being analyzed starting from the point cloud and compare it with the 3D model without defects in order to identify the defective areas. Another popular approach is to analyze the differences between the normals and curvatures of different areas of the point cloud in order to differentiate those that constitute defects, from those that are defect-free.

This work addresses the problem of finding defects and deformations in surfaces represented as 3D point clouds. The aim of the developed algorithm is to find the defective areas on a point cloud, specifying for each point if it belongs to an area that constitutes a defect or not. This is done by exploiting a completely unsupervised approach that is based on an innovative way to exploit the geometry of the point cloud itself. This approach makes use of a technique adopted mainly in the denoising of images, that we have adapted to work in an area still little studied, namely the search for defects in point clouds.

2. Problem formulation

Let's call $P = \{p_i, i = 1, \ldots, I\}$ the input point cloud given to the algorithm. Each point $p_i \in P$ can be defined as:

$$p_i = q_i + \eta_i \qquad p_i, q_i, \eta_i \in R^3$$

where q_i is a point that belongs to the noise-free point cloud Q and $\eta_i \sim \mathcal{N}(\mu, \Sigma)$ is the i.i.d. zero-mean ($\mu = [0,0,0]^T$) Gaussian white noise with $\Sigma = \sigma^2\mathbf{1}$ where $\mathbf{1}$ is the 3x3 identity matrix.

An anomalous point is a point $p_i \in P$ that belongs to a defective region of the point cloud P. Our goal is to define an anomaly score a_i for each point p_i. Each anomaly score can be defined as

$$a_i = \mathcal{A}_P(p_i)$$

where $p_i \in P$ and $\mathcal{A}_P : R^3 \to [0,1]$ is a function that associates a score $a_i \in [0,1]$ to each point p_i of the point cloud P. The anomaly score of each point depends on the point cloud P and, specifically, on the number of points that constitute it.

One of the biggest challenges that we face in this context is the lack of point cloud datasets for damage detection. Most of the available datasets are point clouds that represent structures with defects and malformations, but without labels on the points to identify whether they belong to faultless areas or not. Another element that represents a challenge within this context is the fact that it is not known a priori whether the point cloud given as input to be analyzed actually has damages and defects or not, leading in this way to the necessity of using a completely unsupervised approach.

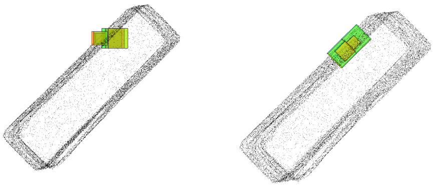

Figure 1: An example of the orientations of the directional neighborhoods of a point p_i. The ones on the left are oriented as the Local Coordinate System and it can be clearly seen that this provides a rough estimate of the surface under the point cloud. On the right side the ideal orientation of the directional neighborhoods.

3. Proposed solution

The proposed model has been developed based on an intuition related to the geometry that characterizes the defects on the surfaces. Defects tend to have a non-uniform and homogeneous geometry and are characterized by sharp and irregular features. The points of the point cloud that constitute the damaged areas are characterized by a high bias, caused by the irregular geometry, and the proposed solution exploits such bias. In particular, the method uses an approach based on the Local Polynomial Approximation and the Intersection of Confidence Intervals (LPA-ICI) rule that allows to generate parallelepipeds called Directional Neighborhoods (DN) that expand on the point cloud depending on the geometry of the surface below the point cloud itself. The DNs that are generated tend to expand as much as possible in areas characterized by points that have a low bias which usually belong to defect-free areas. However, in the proximity of defective areas the DNs assume a rather small size, due to the high bias that characterizes points in such areas. The basic idea of the presented approach is to assign an anomaly score to each point of the point cloud relatively to the number of DNs each point falls into.

Taking as input the noisy point cloud P the first step is to create the Local Coordinate System (LCS) denoted as $\mathcal{L}_i$ for each $p_i \in P$. The axes of this reference system ($x^{\mathcal{L}_i}$, $y^{\mathcal{L}_i}$ and $z^{\mathcal{L}_i}$) are the three principal components c_i, d_i and e_i obtained via PCA considering the K nearest neighbors of p_i. In particular $x^{\mathcal{L}_i} \equiv c_i$, $y^{\mathcal{L}_i} \equiv d_i$

and $z^{\mathcal{L}_i} \equiv e_i$. This allows to create a new reference system that places the currently considered point p_i at the origin and express the relative position of all other points of the point cloud P with respect to this one. The local coordinates in $\mathcal{L}_i$ of each point $p_m \in P$ are denoted as $p_m^{\mathcal{L}_i} = \left[x_m^{\mathcal{L}_i}, y_m^{\mathcal{L}_i}, z_m^{\mathcal{L}_i}\right]^T$, where x_m,y_m and z_m are the coordinates of the point $p_m \in P$ with respect to the original reference system. The relative position $p_m^{\mathcal{L}_i}$ is obtained from the three principal components as

$$p_m^{\mathcal{L}_i} = \begin{bmatrix} c_i \\ d_i \\ e_i \end{bmatrix} (p_m - p_i)$$

The next step is the creation of the DNs. The base size of each DN is $h_j \times h_j$ where $h_j \in R^+$. Let's call H the set that contains such values sorted in increasing order such that
$h_1 < h_2 < \cdots < h_J$.
For each of the four quadrants identified by $x^{\mathcal{L}_i}$ and $y^{\mathcal{L}_i}$ and for each value $h_j \in H$, a DN has to be generated. Its height is equal to the maximum between 6σ, the estimation of the noise of the point cloud, and $2h_j$. In this way it is possible to include in the considered DN those noisy points $\pm 3\sigma$ slightly far from the plane identified by the two main axes of the LCS. Considering that PCA allows to obtain a rough estimate of the surface below the point cloud, the DNs will assume an orientation that is not the ideal one, as shown in Figure 1. This however does not affect the performance of the algorithm. Each DN is identified as $u_{i,\theta}^{h_j}$, where i represents the index of the point in the origin of the LCS, θ the quadrant for which the DN is being computed, and h_j the size of its square base. Each DN $u_{i,\theta}^{h_j}$ has to be intersected with the point cloud expressed in the LCS $\mathcal{L}_i$ in order to obtain the indexes of the points that fall within $u_{i,\theta}^{h_j}$. The set comprising these indices is denoted as $M_{i,\theta}^h$.

The next step is to use the LPA-ICI technique to identify the best DN among the various $u_{i,\theta}^{h_j}$ computed for the p_i point for the θ quadrant. The LPA-ICI technique allows to exploit the computed DNs to obtain from each of them an estimate of the surface underneath the point cloud. This is done by fitting a low-order polynomial model that exploits the points contained in the considered DN. The LPA-ICI allows to expand each DN as much as possible until the points that fall within have a bias too high compared to the assumed polynomial smoothness. Usually this happens in correspondence of sharp features that characterize the defective areas. In this way the DNs that are in correspondence or even within these areas will assume reduced sizes. We are interested in finding the best DNs for each point p_i. To do so the ICI rule requires to compare some estimates obtained through different DNs in order to find the best ones. We have decided to use as estimates for the ICI rule the estimates of the elevation of the surface S underlying the point cloud P at p_i with respect to $\mathcal{L}_i$. For each $k = 1, \ldots, |M_{i,\theta}^h|$ the value of the LPA-kernel $g_{i,\theta}^h$ that allows the fitting of the polynomial model respectively to the DN that is being considered [5] is computed:

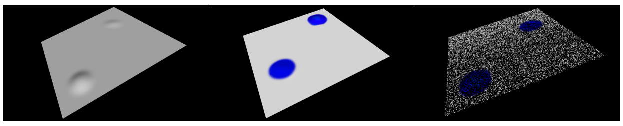

Figure 2: An example of a surface with two dents generated with MeshLab (left), the version where the defects are highlighted with a blue color (center) and the point cloud obtained performing a sampling operation over the colored surface (right)

$$g_{i,\theta}^{h_j}(k) = \phi(k,:) \left(\phi^T \phi\right)^{\dagger} \phi(1,:)^T$$

The polynomial approximation of the normal elevation of the underlying surface S at p_i with respect to $\mathcal{L}_i$ is defined as:

$$\left(\tilde{z}_i^{\mathcal{L}_i}\right)_\theta^{h_j} = \sum_{k=1}^{|M_{i,\theta}^h|} z_{m_k}^{\mathcal{L}_i} g_{i,\theta}^{h_j}(k)$$

In this way, for a particular θ and for each $h_j \in H$, an estimation $(\tilde{z}_i^{\mathcal{L}_i})_\theta^{h_j}$ is identified. The following step is to identify the best size among those in H for each DNs related to the point p_i, one for each quadrant θ. This is done through the ICI rule that makes use of the set of estimates $\{(\tilde{z}_i^{\mathcal{L}_i})_\theta^{h_1}, \ldots, (\tilde{z}_i^{\mathcal{L}_i})_\theta^{h_J}\}$ that have been computed. In particular the pointwise variance of each estimate has to be computed first:

$$\left(\sigma_{i,\theta}^h\right)^2 = \sigma^2 ||g_{i,\theta}^h||_2^2$$

This is necessary to calculate the confidence interval for the ICI rule to identify the best size among those present in H for the currently considered DN. Such confidence interval D_j is defined as:

$$\mathcal{D}_j = \left[\left(\tilde{z}_i^{\mathcal{L}_i} \right)_\theta^{h_j} - \Gamma \sigma_{i,\theta}^{h_j}, \left(\tilde{z}_i^{\mathcal{L}_i} \right)_\theta^{h_j} + \Gamma \sigma_{i,\theta}^{h_j} \right]$$

where $\Gamma \in R^+$ is a threshold parameter that influences the choice of the best size defined by the ICI technique: the higher it is the easier will be to find non empty intersections, allowing the algorithm to choose higher values for the best size. The ICI rule selects as best size h^+ the one among the different sizes $h_j \in H$ that is related to the last interval D_j such that its intersection with the previous intervals is not empty. In this way it is possible to define the DN $u_{i,\theta}^{h^+}$ which has as size h^+, i.e. the best size for the DN of the point p_i in the quadrant θ. This procedure is performed for each point $p_i \in P$. The final step is to compute the anomaly score a_i of each point in the point cloud. To do this we refer to the number of DNs each point falls into. Since the DNs, by construction, can't expand too much in the defective areas, the points in such areas will fall within a few DNs. For this reason the anomaly score of each point $p_i \in P$ is computed as the reciprocal of the number of DNs the point falls into.

4. Experiments

Since the field of point cloud defect detection is quite new and not very deep, it is difficult to find datasets of point clouds representing surfaces with damages and defects. One of the possible options to obtain a point cloud useful for this purpose is to purchase specific instruments, such as laser scanners, and scan surfaces that contain defects and deformations. Given the high cost of this instrumentation we have decided to adopt an approach that allows to obtain point clouds suitable to verify the proper functioning of the developed algorithm, although their quality is not as high as what would be obtained using high precision scanners. The point clouds that have been used in some of the tests are obtained from 3D models to which we have performed a sampling of the surface. In this way it is possible to obtain some point clouds to perform the tests through a procedure very similar to what lasers actually do when they scan real surfaces.

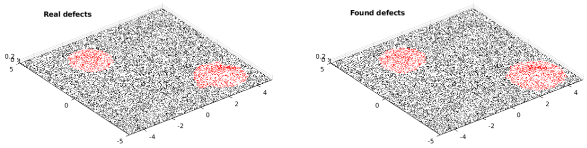

Figure 3: The input point cloud with the real defects colored in red (left) and the output point cloud with the found defects colored in red (right)

In order to analyze the performance of the algorithm on such point clouds it is necessary to refer to some metrics to understand if actually the algorithm is able to recognize adequately the defective areas. The metrics we use are F1-score, AUC, Accuracy, Precision, Recall and False Negative Rate (FNR). In order to compute these metrics it is necessary to define a ground truth: to perform the first tests it has been decided to create 3 very simple 3D models that have defects such as bulges and scratches. Such damages have been colored as shown in Figure 2 in order to be distinguished from the defect-free areas. For these experiments we decided to identify a point as anomalous if its anomaly score was above a certain threshold T. We chose to set this value equal to the value that balances the False Positive Rate and True Positive Rate when defining the ROC curve, i.e. the one closet to the left upper corner of the ROC. Figure 3 shows one of the point clouds obtained from these models with 2 buldges and consisting of 20000 points. The performance results of the algorithm on this point cloud are presented in the Table 1.

F1	AUC	Acc	Prec	Rec	FNR
0.8134	0.9880	0.9517	0.7458	0.8943	0.1057

Table 1: The values of the different metrics for the point cloud represented in Figure 3

From this experiment we can see that the algorithm is able to adequately identify the two defective areas. When searching for defects, it is much riskier to identify as non-defective an area that actually is. For this reason it is fundamental to try to reduce false negatives as much as possible. Therefore particular attention must be paid to the metrics that are most affected by false negatives, namely Recall, FNR and F1. As can be seen from the results regarding these metrics the algorithm performed very well in correctly identifying points belonging to damaged areas.

In addition to this custom point cloud dataset we decided to use 3D models representing civil structures such as walls and columns. This is done in order to verify the performance of the algorithm on models with complex geometry and that are part of the civil sector, in which the search for defects is a very important and delicate operation. Also in this case we obtained the point clouds by sampling the surface of the models. Given their complexity, it is not possible to color them manually to identify the anomalous and damaged areas, as this would led to inaccurate evaluations of the performance of the algorithm. For this reason, not having a ground truth, it is not possible to calculate the metrics used previously, but we relied on a visual comparison between the 3D model and the defects found on its representation as a point cloud by the algorithm.

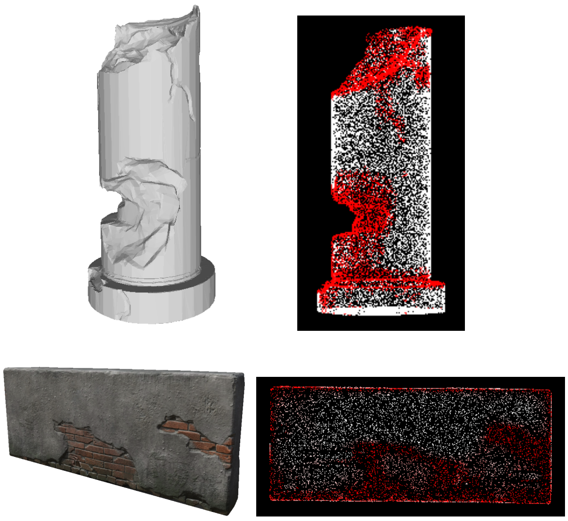

Figure 4: On the top the 3D model of a damaged column (left) and its point cloud representation where the damages detected by the algorithm are colored in red (right). On the bottom, the 3D model of a damaged wall (left) and its point cloud representation where the damages detected by the algorithm are colored in red (right).

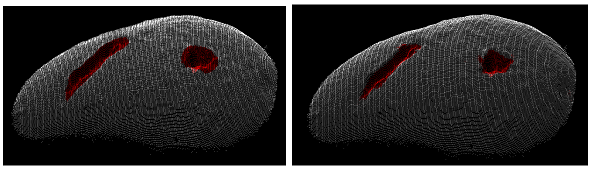

Figure 5: On the left the ground truth of a point cloud of the MVTec 3D-AD dataset and on the right the defects found by the algorithm

The point cloud in on the top right section of Figure 4, consisting of 20000 points, shows how the algorithm is able to adequately identify the main defects of the original 3D model. It can be seen that the algorithm performs well even on a surface such as a column that has some curvature. On the bottom right of Figure 4 we can see the damaged areas identified by the algorithm on a point cloud of 20000 points representing a damaged wall. It can be observed that also in this case the algorithm behaves adequately well in the identification of the defective areas. However, it can also be seen that points belonging to the edges of the model are identified as belonging to damaged areas. This represents a limitation of the algorithm since such areas are characterized by sharp edges in proximity of which the DNs tend to assume reduced dimensions. Therefore, as it happens in the defective areas, the points in these regions fall within a few DNs, making their anomaly score high and therefore identifying them as defects by the algorithm.

Recently the MVTec 3D-AD dataset [2] which contains over 4000 high-resolution scans of 10 different object categories such as potatoes, ropes and tires has been made available. Currently this is the only publicly available dataset created specifically for the task of unsupervised anomaly detection and localization. In their work Bergmann et al. [2] exploits this dataset to test the performance of three methods, namely Voxel f-AnoGAN [4], convolutional Voxel AE [1] and Variation Models [3]. These methods are presented in two variants, with respect to whether they work on voxels or depth images. These performances are used as benchmarks for our algorithm. We analyze 5 point clouds of the category *potato* and, having the ground truth available for each of them, we are able to compute the F1, AUC, Accuracy, Precision, Recall and FNR metrics.

		potato
Voxel	GAN	0.427
	AE	0.484
	VM	**0.652**
Depth	GAN	0.489
	AE	0.549
	VM	0.419

Table 2: The AUC for the baseline methods working with 3D data information.

F1	AUC	Acc	Prec	Rec	FNR
0.8212	**0.8688**	0.9985	0.9260	0.7378	0.2622

Table 3: The values of the different metrics for the point cloud represented in Figure 5

Figure 5 shows one of the 5 point clouds that have been analyzed and Table 3 the results of the algorithm on it. The benchmarks on this category of point clouds are represented in Table 2. Looking at the AUC, we can see that the algorithm outperforms all the baseline methods. This also shows how the algorithm is able to work well even on real point clouds, i.e. obtained through real scanners.

5. Conclusions

In this thesis work we propose an innovative method for finding defects on surfaces represented as point clouds. This completely unsupervised method exploits the LPA-ICI technique that allows to identify damaged areas exploiting the sharp features that characterize defects and deformations. Working in a totally unsupervised way makes the algorithm extremely versatile, enabling its application in any field of application. To operate in a particular domain the algorithm does not require any adaptation on datasets belonging to the considered domain, which are also difficult to find given that this field is still little studied. The tests carried out allow to observe how the algorithm behaves well regardless of the type of point cloud given as input.

Thanks to the MVTec 3D-AD dataset it has been possible to demonstrate how the algorithm works well also on point clouds obtained from real scanners. However the tests carried out on this dataset concern only a small part of the entire set of available point clouds. Consequently a possible future action is to apply the algorithm to the point clouds of all other categories. Moreover it would be useful to apply the algorithm on point clouds of civil structures obtained through specific scanners to verify the effectiveness of the algorithm on real data also coming from this domain.

References

[1] Marcel Bengs, Finn Behrendt, Julia Krüger, Roland Opfer, and Alexander Schlaefer. 3-dimensional deep learning with spatial erasing for unsupervised anomaly segmentation in brain mri, 2021.

[2] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5*, 2022.

[3] Steger C, Ulrich M, and Wideman C. *Machine Vision Algorithms and Applications.* Wiley-VCH, 2 edition, 2018.

[4] Jaime Simarro Viana, Ezequiel de la Rosa, Thijs Vande Vyvere, David Robben, Diana M. Sima, CENTER-TBI Participants, and Investigators. Unsupervised 3d brain anomaly detection. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 133–142. Springer International Publishing, 2021.

[5] Zhongwei Xu and Alessandro Foi. Anisotropic denoising of 3d point clouds by aggregation of multiple surface-adaptive estimates. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):2851–2868, 2021.