www.cad-journal.net
Computer-Aided Design
AND APPLICATIONS

Protrusion-oriented Point Cloud Semantic Segmentation

Alexander Agathos[1] , Philip Azariadis[2]

[1]University of the Aegean, agatha@aegean.gr
[2]University of West Attica, fazariadis@uniwa.gr

Corresponding author: Philip Azariadis, fazariadis@uniwa.gr

Abstract. In recent years the acquisition of point clouds using different 3D sensors (LIDAR, structured light, etc.) or 2D camera sensors using photogrammetry has led to the need to give a higher semantic meaning to the captured raw data so as to allow for a higher-level processing in CAD applications or to use the semantics for various applications like autonomous driving, robotics, urban and rural classifications etc. A way to achieve this is through point cloud semantic segmentation. Point cloud semantic segmentation is the technique to assign to each point of the 3D point cloud a semantic label. A class of point clouds that can be segmented semantically and are of interest in this work are those that are sampled from articulated objects, i.e., having protrusions. This work, extends the work of the protrusion-oriented semantic segmentation of polygon meshes presented in [2] to unstructured point clouds. With the proposed methodology there is no need to reconstruct the polygonal mesh surface of the object in order to perform segmentation since triangulation is a task that can be quite challenging and time consuming, especially with the presence of noise. Detailed experimentation is presented showing that the proposed approach can segment robustly point clouds even in the presence of noise. In terms of computation time the proposed method is comparable to the polygon based approach.

Keywords: Point cloud, semantic segmentation, protrusion function, articulated objects, minimum graph-cut
DOI: https://doi.org/10.14733/cadaps.2024.328-349

1 INTRODUCTION

In recent years point cloud acquisition using either a laser scanner or a camera through photogrammetry has become very popular. This has lead to the collection of a vast number of point clouds leading to the necessity to handle them efficiently, produce variations of them through metamorphosis, search for similar point cloud representations from a database and classify them through object detection for various autonomous applications. One way to tackle these problems is to provide to the point cloud a higher semantic meaning through semantic segmentation.

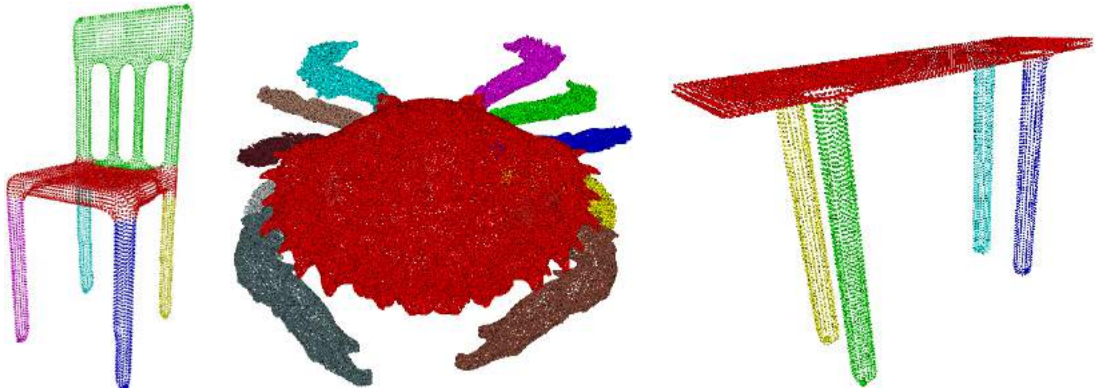


Figure 1: Semantic segmentation of point clouds

Semantic segmentation of a point cloud is the process of assigning to each of its points a semantic label. This label usually classifies the point to a perceptually meaningful part. Point cloud semantic segmentation differs from the known general term point cloud segmentation. In point cloud segmentation the aim is to partition the point cloud into homogenous areas which have a geometric shape like the planar roof and walls of a building, while in semantic segmentation the aim is to give to the segmented parts a characteristic that makes sense in human perception. Examples of the later can be seen in Fig. 1 which shows the classification of all points into their perceptually meaningful parts labelled with different colors.

Semantic segmentation can be unsupervised using various techniques like convexity analysis [17] or using the protrusion function, as it is adopted in this paper. Recently, supervised techniques have been introduced using machine learning techniques like for example Maximum Likelihood Classifiers [19], Support Vector Machines [21], Random Forests [7]. Though the most active field of machine learning in semantic segmentation is nowadays deep learning. With this approach the point cloud is either (i) transformed into multi-view images [33], (ii) transformed into voxels by partitioning it into a volumetric grid [24], and (iii) used directly with its coordinates [28] or with its edge connectivity [39]. In all cases, convolutional neural networks are used to process and generate features. These convolutions are either one-dimensional, two-dimensional or three-dimensional.

The main problem with supervised learning is that there is a need of a significant number of annotated point clouds with labels to train the neural networks, which is cumbersome and needs a lot of human labor. Also, to make things worse, for each new part there is a need to provide new annotations for the network to process it. Of course, a neural network has the ability to generalize so as to be able to label points belonging in a variant of the specified class, but it can't generalize to label a completely new part that it has not been trained to recognize. In contrast, unsupervised semantic segmentation does not have this limitation. It can be used to segment a variety of objects using general geometric characteristics that a wide variety of objects have. A class of point clouds that are of interest in this work is those that are sampled from articulated objects, i.e., having protrusions. This work, extents the work of semantic segmentation of polygonal meshes presented in [2] to unstructured point clouds. With the proposed methodology there is no need to reconstruct the polygonal mesh surface of the object, a task that can be quite challenging and time consuming, especially with the presence of noise.

The rest of the paper is organized as follows. An overview of previous research conducted on point cloud segmentation is presented in Section 2. In Section 3, the proposed methodology for the semantic point cloud-segmentation is presented. Section 4 details the experiments carried out to evaluate the segmentation quality of the proposed approach, and presents an extensive comparison both with the mesh-based counterpart and with a point-based method in the literature. Section 5, presents a discussion highlighting the differences of the proposed point-based method with the mesh-based approach. Finally, Section 6 summarizes the results, and conclusions are drawn.

2 PREVIOUS WORK

According to [40], point cloud segmentation can be divided in methods in which the extracted parts do not have a strong semantic meaning, i.e. they correspond to specific predefined objects or structures in a scene, and those that the parts have strong semantic information. In [40] this semantic information is supposed to be the product of a machine learning process (clustering methods). In this paper, though, it is assumed that a semantic segmentation algorithm is capable to segment the object into semantically meaningful parts, like for example in Fig. 1, the chair object is segmented into its legs, seat and back. If the given application requires an automatic semantic label to be given to the part like foot, leg, arm, head, human, etc. object recognition can be performed based on geometric features (like [29]) or machine learning techniques. In the following both methods will be shortly analyzed. The non-semantic point cloud segmentation is referred as *point cloud segmentation*.

2.1 Point Cloud Segmentation

Point cloud segmentation can be divided in four groups: edge-based, region growing, model fitting, and clustering based.

2.1.1 Edge-based

In edge-based segmentation, first, feature curves are found in the point cloud and then the parts enclosed in these feature curves are extracted. Representative work of this approach is [5] where the feature curves are found when sudden changes in the normal direction occur. The work of [16] further improved the complexity of this approach. The problem with the edge-based approach is its limited applicability since it mostly works on range images where the noise level is low. Also there is no guarantee that the detected feature curves will create closed contours so as to extract the parts belonging inside their boundaries.

2.1.2 Region growing

In the region growing segmentation the most common approach is to start with a seed element like a point in the cloud and grow it by adding neighbours based on various criteria. These criteria are usually geometric features (like normal vectors, curvature) that measure the similarity of the point to be added with the existing region. Besl and Jain [4] describe a two-stage region growing approach in which seed elements are selected based on mean and Gaussian curvature of points of the range-image achieving an initial coarse segmentation. In the second stage region growing further adds point elements guided by bivariate surface fitting. Additional work in region growing to segment building plane structures is reported in [13, 38, 26].

Selecting a single point as a seed element is not ideal for massively large point clouds. In these cases a region of points is preferred as a seed element. An approach to select regions can be achieved by the voxelization of the area of the point cloud. Deschaud et al. [9] present a voxel-based approach in which the seed element is a voxel containing a subset of the point cloud. Vo et al. [38] proposed an octree-based region growing approach for fast initial seed region creation and growing.

The main disadvantage of the region growing approach is that it depends on the criteria for merging which can fail to correctly add elements or lead to oversegmentation.

2.1.3 Model fitting

The basic concept of the model fitting segmentation is to fit basic geometries on subsets of the point cloud like planes, spheres, cylinders, tori, etc. In this way the segmentation of the point cloud is achieved by assigning as segmentation parts the points that fit these primitives. The basic methodologies for model fitting use either the Hough Transform [15, 10] (HT) or RANdom SAmple Consensus [12] (RANSAC).

1. The Hough transform algorithn first creates the parametric representation of the model that needs to be fitted, then it discretizes the parametric representation into a grid and finally the sample points are used for voting for the best fitting parameter tuple on the grid. There has been many subsequent algorithms based on the Hough transform to improve its efficiency like the Probabilistic Hough Transform [18] (PHT), Randomized Hough Transform [41] (RHT) and Kernel-based Hough Transform [11] (KHT). At that time a comparison of all these algorithms for their suitability to segmentation, performed by Bormann et al. [6], revealed that RHT [41] was the best performing algorithm. Later, Limberger et al. [22] extended KHT [11] to 3D point clouds and showed that their approach for segmentation outperforms RHT [41].

2. The Ransac algorithm [12] has two main steps. The first step is hypothesis generation. In this step N random points are chosen and a set of parameters are estimated conforming to the shapes that are expected to be found in the scene, like sphere, planes, cylinders, tori, etc. Second, at the hypothesis evaluation step the most probably hypothesis is chosen, specifically the shape that best fit the point cloud. The advantage of Ransac is that it can perform robustly even when the point-cloud is contaminated with high levels of noise. In the context of Point Cloud Segmentation, RANSAC is used in plane segmentation like building façades [1, 3], building roofs [8] and indoor scenes [20].

2.1.4 Clustering

Clustering is the process of gathering elements into groups with the same homogeneous pattern. In machine learning it belongs in the field of unsupervised learning. This means that the extracted parts usually do not hold a semantic meaning, it is just a group of elements with similar properties. These properties in point cloud segmentation can be geometric features, spectral features or its spatial distribution. The most widely used methods that use clustering for segmentation are K-means and Fuzzy clustering.

1. K-means is the most widely used Clustering method. It separates the point cloud into K regions based on a similarity-metric, which can be the Euclidean, Mahalanobis or Manhattan distance. It has been used for single tree crown segmentation [25] and planar structure extraction from roofs [30].

2. In K-means clustering each point of the point cloud belongs to only one cluster. In Fuzzy clustering each point belongs to all of the K clusters with different membership weights. The point is most probable to belong to the cluster with the maximum weight. As in K-means, the points are clustered using various similarity metrics. Sampath et al. [31] use fuzzy clustering for segmentation and reconstruction of building roofs.

2.2 Point Cloud Semantic Segmentation

In point cloud semantic segmentation the extracted parts needs to have a perceptual meaning in human cognition. The result may be either the product of a machine learning process after training of a model with semantic labels or a method that segments according to how human perception does [14] like in the case of the current work. A recent trend in semantic point cloud segmentation is the use of machine learning methods that use either traditional classifiers or deep learning methods.

2.2.1 Classification of point cloud

These methods use traditional classifiers in machine learning like Gaussian Mixture Models [19], Support Vector Machines [42, 21] and Random Forests [7]. Other methods use statistical approaches like Markov Networks [32], Simplified Markov Random Fields [23] etc. In recent years these traditional machine learning methods have become less popular giving their place to Deep Learning methods.

2.2.2 Deep Learning methods

Deep Learning leverages the classification capabilities of Neural Networks consisting of hundreds of hidden layers, thus the word deep. The most popular Neural Networks for classification and segmentation are the ones which layers perform convolutions (convolutional neural networks). The problem in directly applying these networks to point cloud semantic segmentation is that the point cloud has not a grid structure like an image has. Thus, there was a need for methods able to perform convolution on point clouds. Depending on the data that the Neural Network receive as input the methods that perform segmentation can be categorized to multiview-based, voxel-based and point-based.

1. Multiview-based. In these methods the point cloud is transformed into 2D images taken from different views. Segmentation is performed on these images and the results are combined to produce the final 3D segmentation [33]. These methods were conceived in the early stages of using convolutional neural networks to segment point clouds and are not so popular anymore, since it is difficult to capture the full 3D shape of a complicated object or scene in 2D images.

2. Voxel-based. In these methods the point cloud is transformed into a voxel representation, thus giving it order and structure. On this representation 3D convolutional neural networks can be used to perform segmentation. The segmentation can then be transferred to the point cloud using for example tri-linear interpolation [34]. The main problem of this approach is that the shape of the 3D object is becoming coarse thus there is a possibility to miss features that need to be segmented.

3. Point-based. In these methods the point cloud is not transformed into a different representation but use the point cloud directly. A basic representative of this method is the work of Qi et al. [28] which do not use convolutional neural networks but directly apply multi-layer perceptrons creating augmented features while also performing rigid transformations (rotations, translations) in the training process to capture a general pose of the object. The work Wang et al. [39] use edge connectivity (k-nearest neighbors) to perform convolutions thus capturing the relationships that the points have.

Among the various methods explored above, it is evident that deep learning techniques have gained considerable popularity for point cloud semantic segmentation. However, it is worth noting that these methods necessitate the use of human-made annotations during training. Additionally, a considerable number of samples are required to train the neural network effectively, ensuring that it does not underfit and can generalize accurately. In contrast, the method proposed in this paper can perform semantic segmentation of articulated objects without any training. Moreover, it can also aid in the annotation of training samples for deep learning.

3 PROTRUSION-ORIENTED POINT CLOUD SEMANTIC SEGMENTATION

The work of Agathos et al. [2] segments semantically the triangle mesh of an object containing protrusions. It's extension to point clouds that capture the geometry of an object with protrusions, is not trivial and its constituent steps to achieve a successful segmentation require methods and algorithms suitable for point clouds. The development of such algorithms is a challenging task since, contrarily to polygonal meshes, there is no definition of any kind of surface representing the object. Furthermore, the proposed algorithm is not using

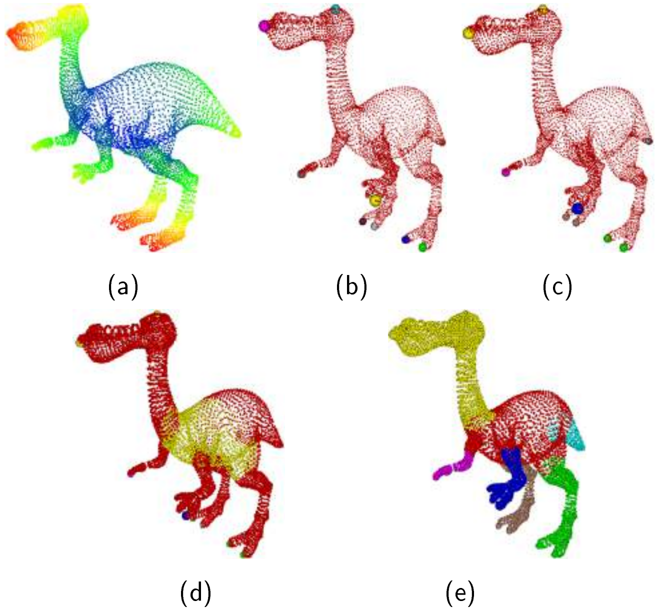


Figure 2: (a) Visualization of the protrusion function calculated on the point cloud, blue to red depict small to large values. (b) The salient points of the object are illustrated with different colors. (c) The salient points are grouped according to the protrusion they belong. (d) The core of the object is illustrated in yellow. (e) The final semantic segmentation of the point cloud is illustrated with each part having a different color.

any other information, like normal vectors, except the raw coordinates of the point cloud. In subsection 3.1 an overview of the steps of the algorithm is given followed by a detailed description of each step in the next subsections.

3.1 Overview of Algorithm

Given a point cloud P consisting of N points $\{p_i\}_{i=1,\dots,N}$ the steps to achieve semantic segmentation are:

(a) Computation of the protrusion function on the point cloud. In this step a function is defined which values distinguish the points on protrusions from the ones that belong near the center of the object, Fig. 2(a).

(b) Computation of the salient points. These points are the extrema of the protrusion function and belong to the protrusions, Fig. 2(b).

(c) Grouping of the salient points. The salient points are grouped according to the protrusion they belong, Fig. 2(c).

(d) Computation of the core of the object. The core (points near the center) of the object is found, Fig. 2(d).

(e) Boundaries and part extraction. The segmentation boundaries are extracted. These are the points that separate the protrusion from the rest of the object. Then the semantic parts are extracted, Fig. 2(e).

Algorithm 1 Compute Geodesic Regions

```
Input: $PointCloud, EGG, thres$
Output: $baseIndexes$, $geodesicRegions$
procedure DivideInGeodesicRegions
    $processedIndexes = \emptyset$
    $remainingPntNumber = N$
    $indexes = \{1, \dots, N\}$
    while $remainingPntNumber > 0$ do
        $neighbors = \emptyset$
        $idx$=a random member of $indexes$
        $indexes- = idx$
        $ProcessedIndexes+ = idx$
        $neighbors+ = idx$
        $baseIndexes+ = idx$
        $neighborhood = GeodesicNeighborhood(PointCloud, idx, EGG, thres)$
        for $n$ in $neighborhood$ do
            if $n$ not in $processedIndexes$ then
                $indexes- = n$
                $processedIndexes+ = n$
                $neighbors+ = n$
                $remainingPntNumber - -$
        $geodesicRegions+ = neighbors$
```

3.2 Computation of the Protrusion Function

The protrusion function $f_{prot} : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined for each point p of the point cloud P as:

$$f_{prot}(p) = \sum_{p' \in P} g(p, p') \tag{1}$$

where $g(p, p')$ the geodesic distance of point p to point p'. The geodesic distance is found by applying the Dijkstra algorithm [35] using the Elliptic Gabriel Graph [27] to provide connectivity between the raw points. The protrusion function receives small values at the center of the object and high values at its extrema (protrusions), see Fig. 2(a).

The problem with the computation of the protrusion function is its high complexity since it has to be calculated for each point of the point cloud making it intractable to point clouds consisting of thousands of points. In [2] an approximation to the protrusion function is proposed and it is followed also in this work. Specifically, the point cloud is divided into non overlapping geodesic regions. These regions are created by applying the Dijkstra algorithm constrained with a distance threshold on randomly chosen points of the point cloud.

To ensure consistency and comparability across all point clouds, it is assumed that the point cloud has been scaled to fit within the unit cube. This scaling facilitates the use of threshold values mentioned in this paper, which remain consistent and constant regardless of the point cloud being segmented.

In Algorithm 1 the aforementioned procedure to divide the point cloud in geodesic regions is shown in pseudo-code. It receives as input the point cloud ($PointCloud$), the elliptic Gabriel graph (EGG) and the threshold value ($thres$) required for the creation of the geodesic neighborhood. After its completion the output will consist of the indexes of the centers of the geodesic regions ($baseIndexes$) and the indexes of the point

Algorithm 2 Compute the Geodesic Neighborhood

Input: $idx, PointCloud, EGG, thres$
Output: $geodesicNeighborhood$

```
procedure GeodesicNeighborhood
    $g = array(N, +\infty)$
    $geodesicHeap = \emptyset$
    $geodesicElement = \{idx, 0.0\}$
    $g(idx) = 0.0$
    $geodesicHeap += geodesicElement$
    while $geodesicHeap$ not empty do
        $geodesicElement = geodesicHeap$ top element and extract it
        $v = index(geodesicElement)$
        $neighbors = EGG(v)$
        for $v_a$ in $neighbors$ do
            $dist = g(v) + ||PointCloud(v) - PointCloud(v_a)||$
            if $dist < thres$ and $dist < g(v_a)$ then
                $geodesicElement = \{v_a, dist\}$
                $geodesicHeap += geodesicElement$
                $g[v_a] = dist$
    $geodesicNeighborhood = \emptyset$
    for $idx = 0$ to $N - 1$ do
        if $g(idx) < +\infty$ then
            $geodesicNeighborhood += idx$
```

cloud contained in each region ($geodesicRegions$). In this procedure a random index (idx) is chosen from the point cloud and it is set as a base index. Then the geodesic neighborhood around this point which radius is smaller than $thres$ is found. From these neighbors a new region is created containing the points that do not belong in other regions.

In Algorithm 2 the procedure to calculate the indexes of a geodesic neighborhood is shown in pseudo-code. It receives as input the center index (idx), the point cloud ($PointCloud$), the elliptic Gabriel graph (EGG) and the required threshold ($thres$) to constrain its extent. It returns the indexes of the geodesic neighborhood ($geodesicNeighborhood$). In the algorithm, a priority queue (heap) is used to store the elements already visited having as key the geodesic distance. Elements are extracted from and added in the priority queue as long as the geodesic distance is smaller than the threshold set. Finally all indexes of the point cloud whose geodesic distance is smaller than $+\infty$ are added in the geodesic neighborhood.

Having partitioned the point cloud into N_R geodesic regions the protrusion function can be approximated as:

$$f_{prot}(p) \simeq \sum_{i \in baseIndexes} g(p, p_i) area(geodesicRegions(i)) \tag{2}$$

where $area(geodesicRegions(i)$ is the area of the geodesic region with center p_i. It is computed by finding the mean-radius of the 1-ring neighborhood of each point contained in the region and taking the sum of the disk areas defined by these radii.

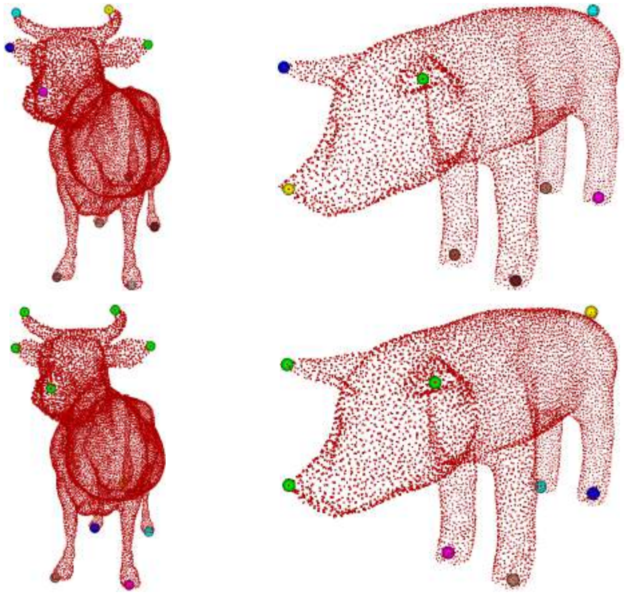


Figure 3: Top row, the salient points found on a cow and pig model. Bottom row, the salient points are grouped according to the protrusion to which they belong.

3.3 Computation and Grouping of the Salient Points

The salient points are points that belong to the protrusions of the point cloud, where the protrusion function receives a local maximum. A point v of the point cloud is considered to belong on a protrusion of the point cloud if its protrusion function is greater than a threshold, i.e. $f_{prot}(v) > t_{prot}$. A point v_s belonging to a protrusion, is also a salient point if its protrusion function for a neighborhood P_{v_s} around it receives a local maximum, i.e. $f_{prot}(v_s) > f_{prot}(v), v \in P_{v_s}$. The neighborhood P_{v_s} can be either a small geodesic neighborhood created by Algorithm 2 with a small value of threshold (e.g. $thres < 10^{-3}$) or a k-ring neighborhood which can be created from the connectivity of the elliptic Gabriel graph. Fig. 3, top row, shows the salient points found on a cow and pig model following the aforementioned methodology.

As it can be observed on the top row of Fig. 3, the protrusion corresponding to the head of the cow and pig model has more than one salient points. These salient points represent the same protrusion and should be grouped together. Assuming the set of salient points is defined as $S = \{s_i, i = 1...N_s\}$, a value T_s is defined:

$$T_s = \frac{\sum_{i=1}^{Ns-1} \sum_{j=i+1}^{Ns} g(s_i, s_j)}{N_s(N_s-1)} \tag{3}$$

where $g(s_i, s_j)$ denote the geodesic distance of the salient points s_i and s_j.

A group of salient points C contains the salient points whose geodesic distance with each other is smaller than the threshold value T_s and is defined as:

$$C = \{s_i \in S : \forall s_j \in C, g(s_i, s_j) < T_s\} \tag{4}$$

Fig. 3, bottom row, shows the grouping of the salient points for the cow and pig models. All salient points that belong to the heads are grouped into the same cluster.

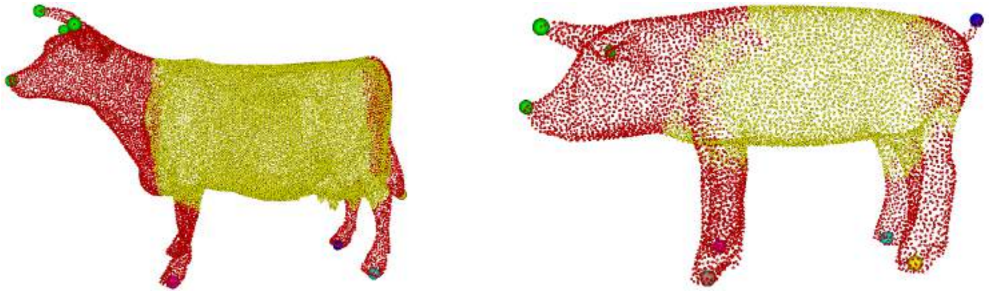


Figure 4: The cores of the cow and pig models are illustrated with yellow color

The *representative salient point* of each protrusion, is the salient point of the group C belonging to the protrusion with the highest protrusion value, i.e.

$$Rep(C) = \{s_i \in C : f_{prot}(s_i) >_{prot} (s_j), \forall s_j \in C\} \tag{5}$$

3.4 Computation of the Core of the Object

A crucial step of the segmentation process is the extraction of the core of the object. By core extraction, it is meant the approximation of the main body of the object. An algorithm which approximates the main body of the object is the one that can acquire all the points of the point cloud except those that belong to the protrusions of the object, separating the later from each other.

The core approximation is performed by using the minimum cost paths between the representative salient points, see Eq. (5). It is assumed that $\hat{S} = \{\hat{s}_i, i = 1, \dots N_c\}$ is the set of representative salient points, where N_C denote the number of clusters found in subsection 3.3 and $\hat{s}_i$ the representative of the i^{th} group of salient points. Also, let $P_a = \{P_{ij}, i, j \in \{1, \dots, N_C\}\}$ be the set of all minimum cost paths of the points of $\hat{S}$, where P_{ij} denote the minimum cost path between $\hat{s}_i$, $\hat{s}_j$. The minimum cost paths are found by backtracking from the destination to the source salient point after applying the Dijkstra algorithm on all representatives using the elliptic Gabriel graph for connectivity between the points of the point cloud.

The central concept of this algorithm revolves around the expansion of a point set in ascending order of their protrusion function values. This expansion continues until the enlarged set encompasses a predetermined percentage of all elements within P_a. By employing this method of point set expansion, the algorithm ensures that it not only reaches the protrusions but also covers an area contained within them. Additionally, a threshold, denoted as t_c, is defined within the algorithm, which represents the percentage of minimum cost path coverage.

In Fig. 4 the cores of the cow and pig models are illustrated. It can be observed that in both cases the core spreads from the center of the objects to their protrusions separating them.

3.5 Boundary extraction of the protrusions

The boundary of a protrusion separates it from the main body of the object. It is considered that between the boundary and the main body of the object an abrupt change in the area happens. The goal is to detect this change, mark this region as the protrusion boundary, and extract the points of the point cloud belonging to it. To achieve this, closed successive regions of points from the point cloud are constructed using a distance function $D_{\hat{s}}$ associated to the salient point representative $\hat{s}$ of the protrusion. The abrupt change that marks the protrusion boundary is detected by examining the ratio of the areas of successive regions. Fig. 5 illustrates

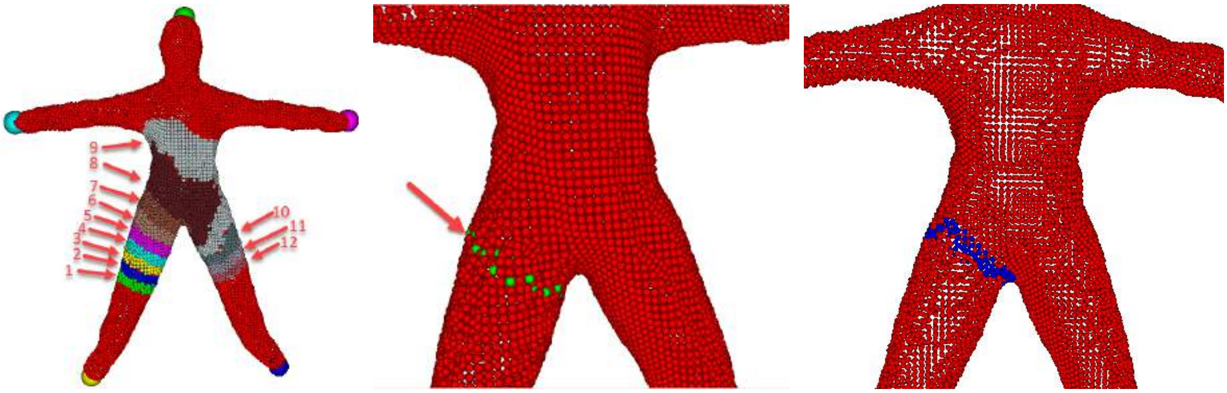


Figure 5: Left, successive regions constructed shown with different colors, the abrupt change in area happens in the vicinity of the red arrow. Middle, the green points mark the segmentation boundary. Right, the graph cut refines the segmentation boundary and passes through deep concavities.

the successive regions constructed on the left, and the points that mark the abrupt change in area are shown in the middle.

In the following, the definition of the distance function $D_{\hat{s}}$ that creates the closed regions and the procedure to place the closed regions of points on the protrusion is presented.

For a specific representative salient point $\hat{s}$ that represents a protrusion, the distance function $D_{\hat{s}}$ for each point v of the point cloud P is defined as the shortest distance between $\hat{s}$ and v. The shortest distance is computed using the Dijkstra algorithm with source $\hat{s}$ and connectivity defined by the elliptic Gabriel graph. The cost for each edge (u, v) of the elliptic Gabriel graph is defined as:

$$cost(u,v) = \delta \frac{\|u-v\|}{average\_dist} + (1-\delta)\frac{|f_{prot}(u) - f_{prot}(v)|}{average\_prot} \tag{6}$$

where $\delta \in [0,1]$, $average\_dist$ and $average\_prot$ the average values of all pair differences in length and protrusion, respectively. A proper balance between the two terms of Eq. (6) creates a closed region of points as it is shown in Fig. 5(left). In this work, the value of δ is set to 0.4.

Let v_{min} denote the point belonging to the core of the object with the minimum geodesic distance from $\hat{s}$ and let $D_{coremin} = D_{\hat{s}}(v_{min})$. Let P_D be the subset of P where each point v has distance function in the range $D_{\hat{s}}(v) \in [(1-d_1)D_{coremin}, (1-d_2)D_{coremin})]$, where d_1 and d_2 denote the extents of this subset. In Fig. 5(left) the union of all the coloured regions illustrate the aforementioned point cloud subset. The successive regions of points should be placed in this subset. Let l_{reg} be the number of regions that the set P_D is divided. The successive regions will have width $\varepsilon = \frac{(\delta_1+\delta_2)D_{coremin}}{l_{reg}}$. It is trivial to extract the points that belong in this width with proper thresholding. Fig. 5(left) illustrates the construction of twelve successive regions on the human feet with different colors. The regions start from the bottom of the right foot, pass from the core, and end up to the left foot of the human point cloud.

The search for the area where the protrusion boundary belongs is performed iteratively by examining the ratio of the areas between successive regions which points belong in the interval $[(1-d_1)D_{coremin}, (1-d_1)D_{coremin} + i * \epsilon]$ and $[(1-d_1)D_{coremin}, (1-d_1)D_{coremin} + (i+1) * \epsilon]$, $i = \{1, \ldots l_{per} - 1\}$. Let $area_i$ and $area_{i+1}$ be these areas respectively. The following ratio r_i is defined:

$$r_i = \begin{cases} \frac{area_{i+1}}{area_i} & \text{if } area_{i+1} > area_i \\ \frac{area_i}{area_{i+1}} & \text{if } area_{i+1} \leq area_i \end{cases}, i = 1, \ldots, l_{per} - 1 \tag{7}$$

The search terminates when there is a r_k greater than a threshold r_{max}. When this happens this region is considered to contain the segmentation boundary. This region is thinned with thresholding providing the final segmentation boundary, see Fig. 5(middle).

In order to avoid creating skewed segmentation boundaries the representative salient point is refined before searching for the segmentation boundary. The procedure to do this starts by finding the point p_{min} of the point cloud with the minimum geodesic distance from all salient points belonging to the group. Then the point c_{min} belonging to the core with minimum geodesic distance to p_{min} is found. Let d_{min} denote the geodesic distance between p_{min} and c_{min}. Then the first point p_{thres} belonging to the geodesic path connecting p_{min} to c_{min}, starting from p_{min}, with geodesic distance greater than $0.3d_{min}$ is selected. Then the region containing the points with protrusion function greater than $f_{prot}(p_{thres})$ is selected. The protrusion function constrained in this region is found and the point with the smallest protrusion function is regarded as the refined salient representative. For further details see [2], Section 3.4.

3.6 Segmentation Boundary Refinement

As can be seen in Fig. 5 the segmentation boundary does not entirely reside where human perception would place it. Hoffman and Singh [14] describe how human perception place the segmentation boundaries on salient parts (protrusions). Specifically the segmentation boundaries should reside at highly concave areas. In order to find highly concave areas in a point cloud the jet-fitting function of CGAL [36] is used. This function fits to each point of the point cloud a polynomial from which the principal curvatures are computed. This fitting is also quite robust in cases where the point cloud is contaminated with noise. From the principal curvatures the Gaussian curvature is computed. Deep concavities reside in areas where the Gaussian curvature receive large negative values.

As in [2] a minimum cut approach will be followed to place the segmentation boundaries. The flow network graph is defined by the elliptic Gabriel graph and the edge-capacities required for the graph cut algorithm will be defined on the midpoints of each of its edges. Specifically for each edge $e = (u, v)$ of the elliptic Gabriel graph the middle point p_e is found by averaging the points of the point cloud corresponding to its endpoints. The Gaussian curvature of p_e is denoted as G_e.

Three subsets of the point cloud are defined. Set **A** contains all points of the examined protrusion, set **C** contains the points in which the segmentation boundary is contained, and set **B** contains the rest of the points of the point cloud. Let $\mathbf{G_c} = (\mathbf{V_C}, \mathbf{E_C})$ be the subset of the graph defined by the points of **C** and the edges of the Gabriel graph which endpoints belong to **C**. Also, let $\mathbf{V_{CA}}$ be the points of **A** which have at least one point in **C** as a 1-ring neighbor and let $\mathbf{V_{CB}}$ be the points of **B** which have at least one point in **C** as a 1-ring neighbor. The flow network graph **G** = (**V**, **E**) is constructed by defining also two more points s, t with **V**, **E** defined as:

$$\begin{aligned} \mathbf{V} &= \mathbf{V_C} \cup \mathbf{V_{CA}} \cup \mathbf{V_{CB}} \cup \{s, t\} \\ \mathbf{E} &= \mathbf{Ec} \cup \{(s, v), \forall v \in \mathbf{V_{CA}}\} \cup \{(t, v), \forall v \in \mathbf{V_{CB}}\} \cup \\ & \quad \{e_{uv} \in EGG : u \in \mathbf{V_C}, v \in \{\mathbf{V_{CA}} \cup \mathbf{V_{CB}}\}\} \end{aligned} \tag{8}$$

Assuming that on each edge of the elliptic Gabriel Graph the Gaussian value G_e is normalized in [0,1] and it is denoted by $\hat{G}_e$, the capacity of each edge $e \in \mathbf{E}$ is defined as $Cap(e) = n * \hat{G}_e$, where $n = 1$ when $G_e > 0$ and $n << 1$ when $G_e < 0$.

Fig. 5(right) shows in blue where the minimum-cut partitions the leg extrusion. It can be observed that it is located in the area where human perception would place the segmentation boundary.

Figure 6: Segmentation of various models with the mesh-based approach vs the proposed approach with and without noise.

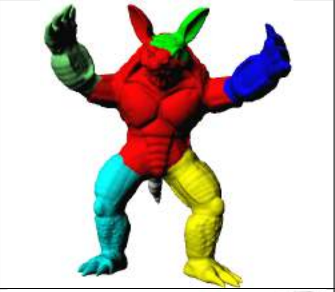
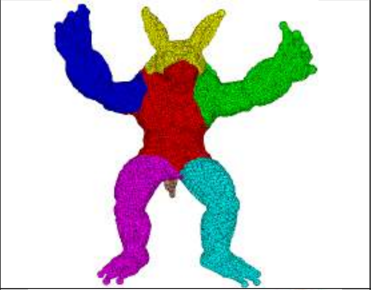
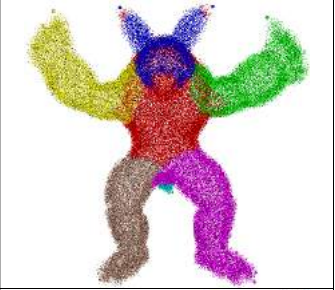
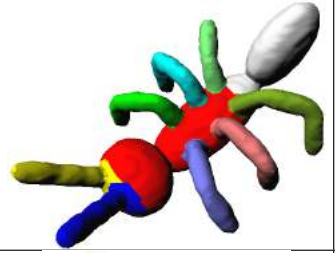
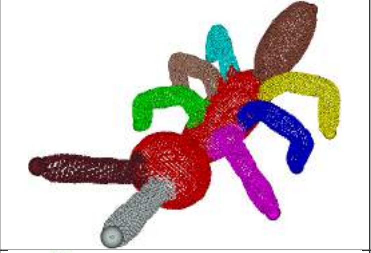
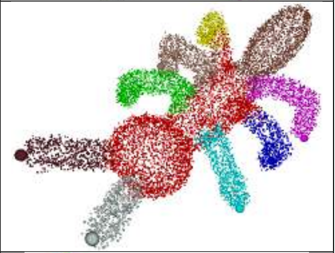
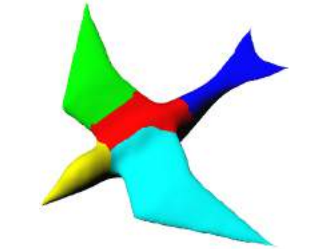
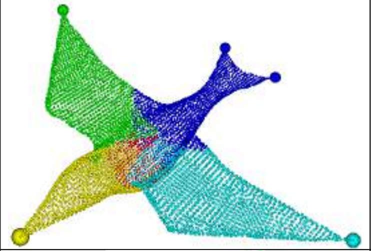
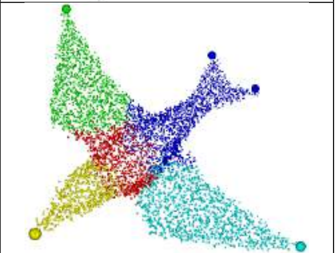
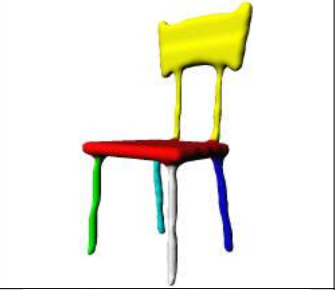
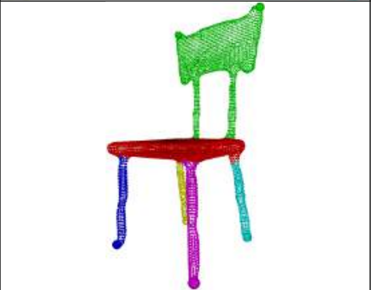
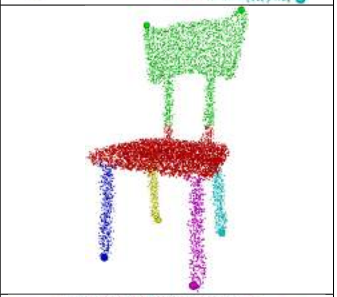
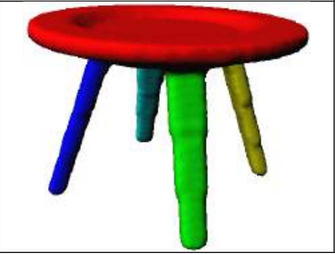
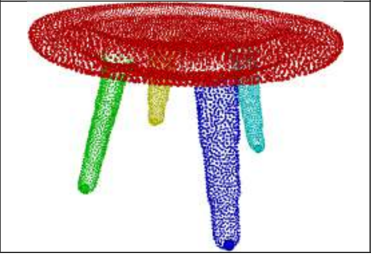
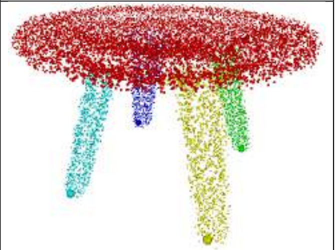
Model	mesh-based	point-based	point-based with noise
Armadillo			
Ant			
Bird			
Chair			
Table			

Figure 6: Segmentation of various models with the mesh-based approach vs the proposed approach with and without noise.

Continued from previous page.

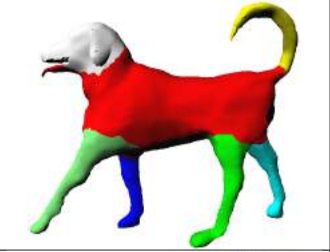
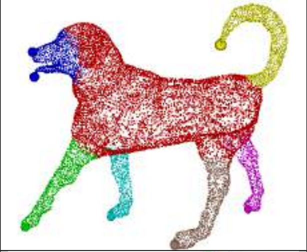
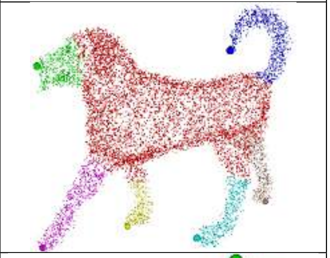
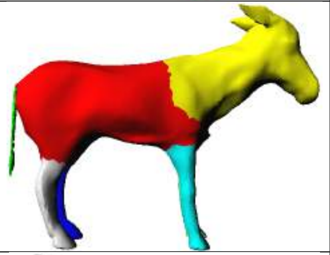
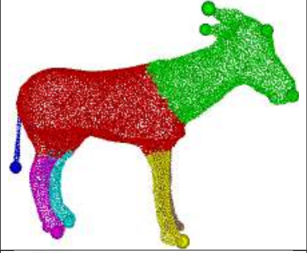
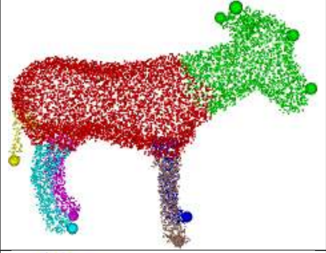
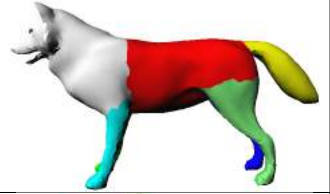
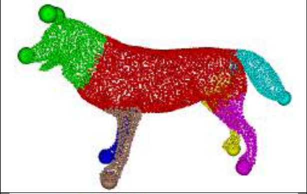
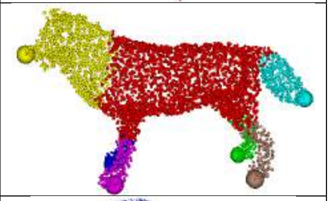
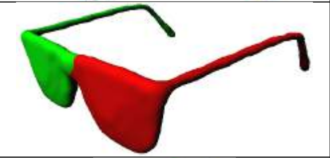
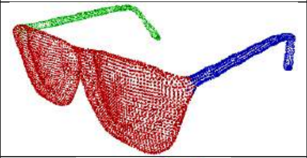
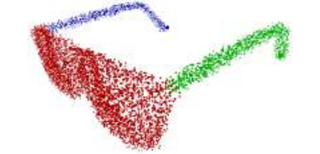
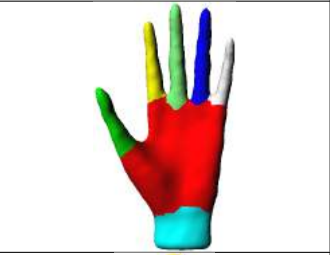
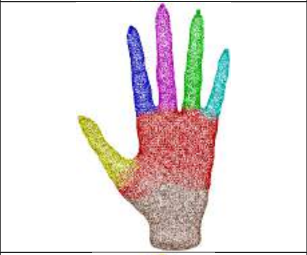
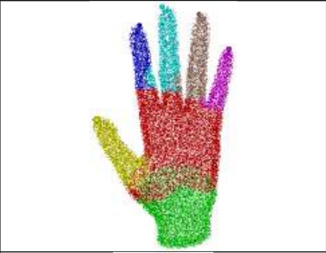
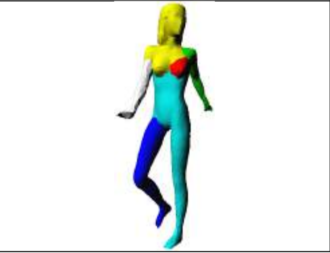
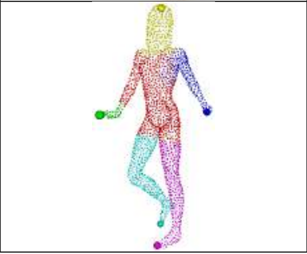
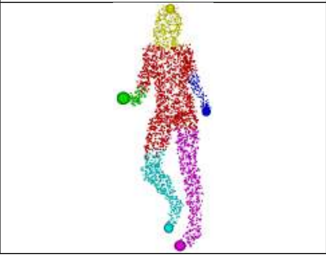
Model	**mesh-based**	**point-based**	**point-based with noise**
Dog			
Donkey			
Fox			
Glasses			
Hand			
Human			

Figure 6: Segmentation of various models with the mesh-based approach vs the proposed approach with and without noise.

Continued from previous page.

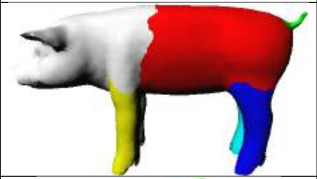
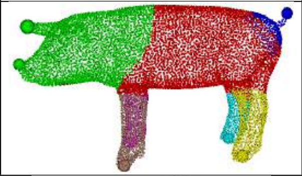
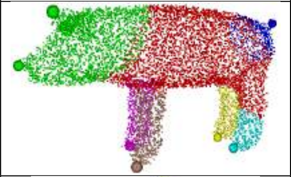
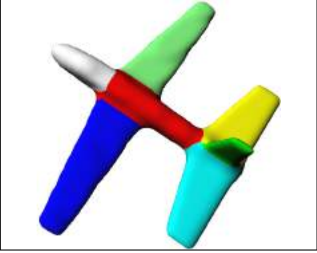
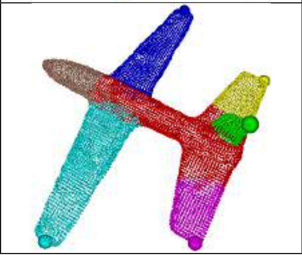
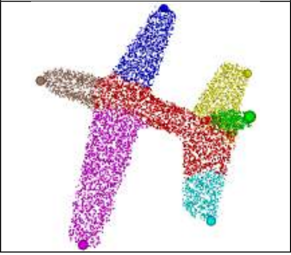
Model	mesh-based	point-based	point-based with noise
Pig			
Plane			

Figure 7: Segmentation of the models of Fig. 6 with WCSeg vs the proposed approach.

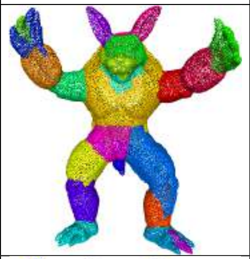
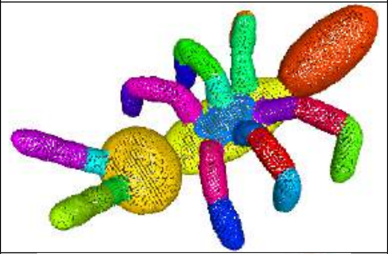
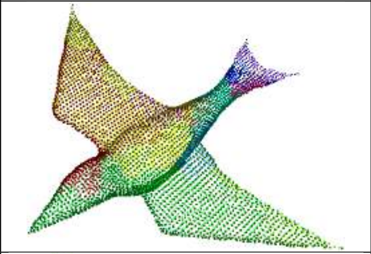
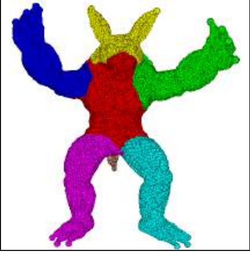
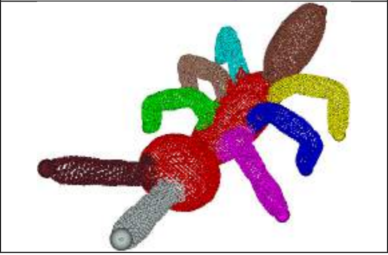
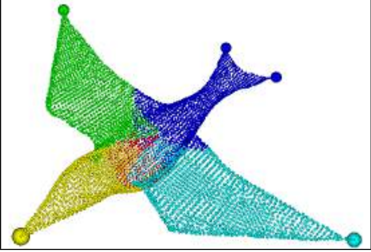
Method	Armadillo	Ant	Bird
WCSeg			
Point-based			

Figure 7: Segmentation of the models of Fig. 6 with WCSeg vs the proposed approach.

Continued from previous page.

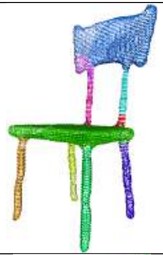
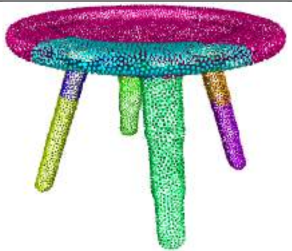
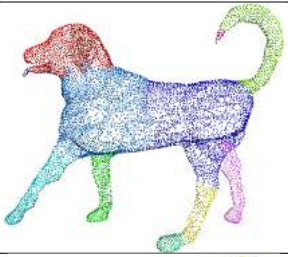
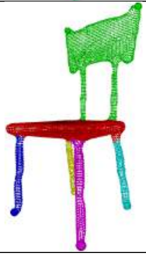
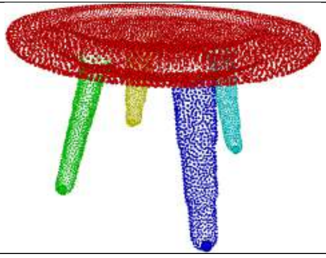
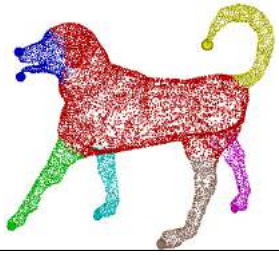
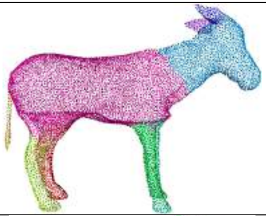
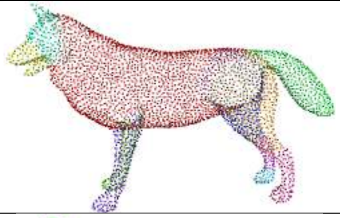
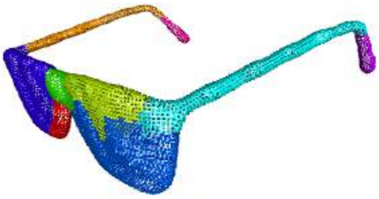
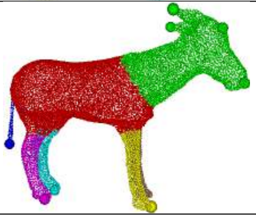
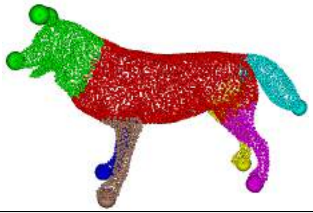
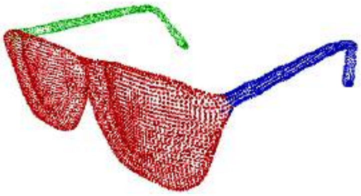
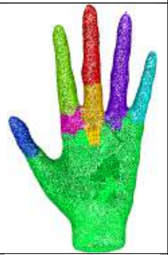
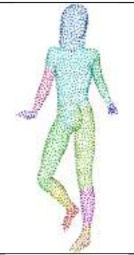
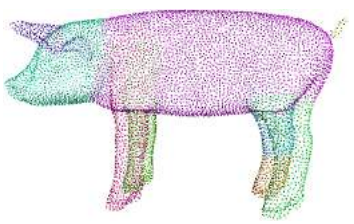
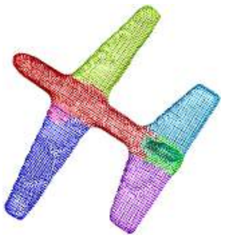
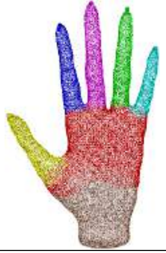
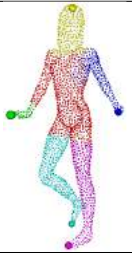
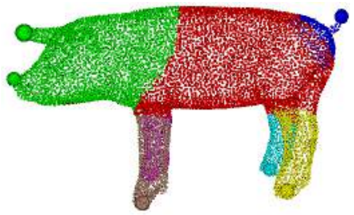
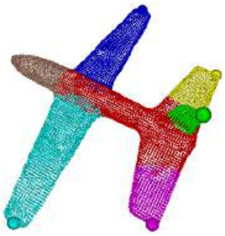
Method	**Chair**	**Table**	**Dog**
WCSeg			
Point-based			
Method	**Donkey**	**Fox**	**Glasses**
WCSeg			
Point-based			

Figure 7: Segmentation of the models of Fig. 6 with WCSeg vs the proposed approach.

Continued from previous page.

Method	Hand	Human	Pig	Plane
WCSeg				
Point-based				

4 EXPERIMENTS

In this section the segmentation capabilities of the proposed algorithm are examined. First, the proposed algorithm is compared with its mesh-based counterpart [2]. Next, the segmentation consistency of the algorithm is tested when the point cloud is contaminated with noise. Then our method is compared against the WCSeg method reported in [17]. Finally, the computation time to perform semantic segmentation is compared with the mesh-based approach. In all the experiments the parameters are set to $t_c = 0.15$, $\delta = 0.4, d_1 = 0.3, d_2 = 0.4, l_{per} = 12$ and $r_{max} = 1.3$. It has been found experimentally that these parameters produce the best results for a broad range of articulated objects.

In Fig. 6 (first and second column) the proposed algorithm is compared with the mesh-based counterpart. It is clearly shown that both algorithms are consistent into the segmentation output managing to extract similar semantic parts. In the third column, the point cloud is contaminated with noise; specifically the points of the noise-free models are randomly displaced with a maximum distance of 1 percent from their original position. It can be observed that despite of this heavy noise, the segmentation is consistent with the noise-free segmentation. The differences between the mesh-based segmentation and the point-based segmentation are due to the fact that the two methods, although similar in spirit, operate on two entirely different domains. Also, the underneath algorithms are very different and therefore it is impossible to produce identical results. However, it is evident that the point-based method proposed in this paper has the capacity to produce consistent segmentation results which are in several cases closer to the human perception than those of the mesh-based approach (see e.g., Armadillo, Ant, Donkey, Fox, Human).

In Fig. 7 the comparison of our algorithm with [17], named as WCSeg, is illustrated. In this work, the point cloud is decomposed into approximate convex components, which are then merged into consistent parts based on a non-local geometric signature. The code is available in [37]. From the given results it can be observed that the approach proposed in this paper segments the point clouds more consistently.

It is worth noting that both the mesh-based approach and the WCSeg method have been unable to provide valid semantic segmentations when the point cloud (or mesh data) is contaminated with noise. Therefore, we are unable to provide any results from these algorithms in such cases.

In Table 1 the execution time of the proposed point-based approach vs the mesh-based approach is listed, for all models of Fig. 6. The calculations were performed on the same processor (Intel I5-11600K with 12 threads) taking advantage its multi-core capabilities with OpenMP. It can be observed that the execution time of the proposed point-based approach is competitive with the mesh-based approach.

Table 1: Execution time of the proposed point-based vs the mesh-based approach (in seconds).

Model	**#Points**	**#Triangles**	**mesh-based time**	**point-based time**
Armadillo	172974	345944	13.308	8.220
Ant	14800	29596	0.931	1.038
Bird	5986	11968	0.558	0.650
Chair	9153	18306	0.681	0.309
Table	13714	27424	0.642	0.580
Dog	9328	18652	0.933	0.430
Donkey	10436	20868	0.834	0.435
Fox	4712	9420	0.557	0.270
Glasses	7407	14810	0.272	0.426
Hand	43442	86880	2.928	2.456
Human	2639	5274	0.348	0.176
Pig	8411	16818	0.768	0.335
Airplane	7351	14698	0.376	0.440

5 DISCUSSION

Given the extensive experiments detailed in Sec. 4, we are able to summarize the advantages of the proposed point-based method over the mesh-based approach as follows: (i) The new method eliminates the need for triangulation of the point cloud before segmentation - a process that can be complex and error-prone, especially when noise is present. (ii) The new method is capable of handling point clouds with complex shapes and significant deviations in the sampling ratio, see Fig. 8. Creating polygonal meshes from such point clouds can often be a challenging task. (iii) The new method demonstrates consistency in handling noisy point clouds. In contrast, mesh-based approaches typically necessitate rigorous preprocessing to filter out noise and ensure a high-quality triangulation prior to segmentation.

On the other hand, the proposed point-based approach needs a sufficient sampling so as to create proper connectivity between the points of the point cloud. Also, protrusions that are very close to the main body can be miscalculated and connected with the main body (or other parts) due to the absence of connectivity between the points.

6 CONCLUSIONS

In this work, a new methodology for the semantic segmentation of point clouds has been presented based on the main workflow implemented for polygonal meshes by Agathos et al. [2]. New algorithms have been devised for each step of the methodology which are applicable to point clouds. With the new methodology a point

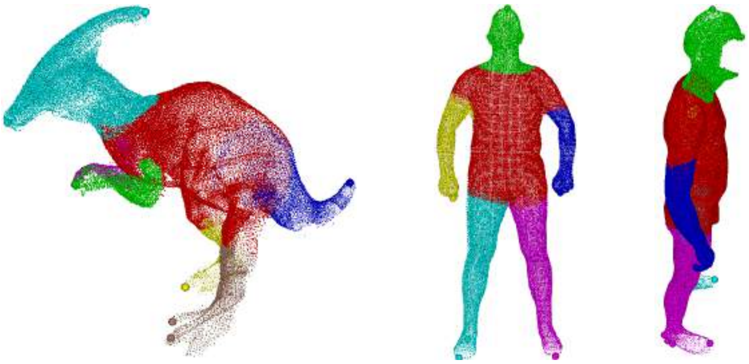


Figure 8: Segmentation of two complex point-clouds with varying sampling and missing information.

cloud of an articulated object can be successfully segmented to semantic parts without the need to produce a polygonal mesh which can be a tedious process in cases of complicated topology and/or noise. Also, the new methodology does not need the normals of the points of the point cloud making it more resistant to noise. This work has an advantage over machine learning methods because it can segment a vast variety of articulated objects, as Fig. 6 and 7 show, without the need of preprocessing as in neural networks. Also in terms of computational time the proposed segmentation is as fast as the mesh-based segmentation of [2].

Acknowledgements

This research has been financially supported by the Initiative "Regional Excellence", under the Operational Program "Competitiveness, Entrepreneurship & Innovation" (EPAnEK), which is co-funded by Greece and European Union (MIS 5047046).

REFERENCES

[1] Adam, A.; Chatzilari, E.; Nikolopoulos, S.; Kompatsiaris, I.: Hransac: A hybrid point cloud segmentation combining 2d and 3d data. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4(2), 2018. http://doi.org/10.5194/isprs-annals-iv-2-1-2018.

[2] Agathos, A.; Pratikakis, I.; Perantonis, S.J.; Sapidis, N.S.: Protrusion-oriented 3d mesh segmentation. Vis. Comput., 26(1), 63–81, 2010. http://doi.org/10.1007/s00371-009-0383-8.

[3] Bauer, J.; Karner, K.; Schindler, K.; Klaus, A.; Zach, C.: Segmentation of building from dense 3d point-clouds. In Proceedings of the ISPRS. Workshop Laser Scanning Enschede, 12–14, 2005.

[4] Besl, P.J.; Jain, R.C.: Segmentation through variable-order surface fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(2), 167–192, 1988. http://doi.org/10.1109/34.3881.

[5] Bhanu, B.; Lee, S.; Ho, C.C.; Henderson, T.: Range data processing: Representation of surfaces by edges. In Proceedings of the Eighth International Conference on Pattern Recognition, 236–238. IEEE Computer Society Press, 1986.

[6] Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A.: The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. 3D Research, 2(2), 3, 2011. http://doi.org/10.1007/3dres.02(2011)3.

[7] Chehata, N.; Guo, L.; Mallet, C.: Airborne lidar feature selection for urban classification using random

forests. In International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 38, 207–212, 2009.

[8] Chen, D.; Zhang, L.; Mathiopoulos, P.T.; Huang, X.: A methodology for automated segmentation and reconstruction of urban 3-d buildings from als point clouds. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(10), 4199–4217, 2014. http://doi.org/10.1109/jstars.2014.2349003.

[9] Deschaud, J.E.; Goulette, F.: A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In 3DPVT, 2010.

[10] Duda, R.O.; Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11–15, 1972. http://doi.org/10.1145/361237.361242.

[11] Fernandes, L.A.; Oliveira, M.M.: Real-time line detection through an improved hough transform voting scheme. Pattern Recognition, 41(1), 299–314, 2008. http://doi.org/10.1016/j.patcog.2007.04.003.

[12] Fischler, M.A.; Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395, 1981. http://doi.org/10.1016/b978-0-08-051581-6.50070-2.

[13] Gorte, B.: Segmentation of tin-structured surface models. In International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, vol. 34, 465–469, 2002.

[14] Hoffman, D.D.; Singh, M.: Salience of visual parts. Cognition, 63(1), 29–78, 1997. ISSN 0010-0277. http://doi.org/https://doi.org/10.1016/S0010-0277(96)00791-3.

[15] Hough, P.V.: Method and means for recognizing complex patterns, 1962.

[16] Jiang, X.Y.; Meier, U.; Bunke, H.: Fast range image segmentation using high-level segmentation primitives. In Proceedings Third IEEE Workshop on Applications of Computer Vision, 83–88. IEEE, 1996.

[17] Kaick, O.V.; Fish, N.; Kleiman, Y.; Asafi, S.; Cohen-OR, D.: Shape segmentation by approximate convexity analysis. Transactions on Graphics, 34, 1–11, 2014. ISSN 0730-0301. http://doi.org/10.1145/2611811.

[18] Kiryati, N.; Eldar, Y.; Bruckstein, A.M.: A probabilistic hough transform. Pattern Recognition, 24(4), 303–316, 1991. http://doi.org/10.1016/0031-3203(91)90073-e.

[19] Lalonde, J.; Unnikrishnan, R.; Vandapel, N.; Hebert, M.: Scale selection for classification of point-sampled 3-d surfaces. In Fifth International Conference on 3D Digital Imaging and Modeling (3DIM 2005), 13-16 June 2005, Ottawa, Ontario, Canada, 285–292. IEEE Computer Society, 2005. http://doi.org/10.1109/3DIM.2005.71.

[20] Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L.: An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells. Remote Sensing, 9(5), 433, 2017. http://doi.org/10.3390/rs9050433.

[21] Li, Z.; Zhang, L.; Tong, X.; Du, B.; Wang, Y.; Zhang, L.; Zhang, Z.; Liu, H.; Mei, J.; Xing, X.; Mathiopoulos, P.T.: A three-step approach for TLS point cloud classification. IEEE Trans. Geosci. Remote. Sens., 54(9), 5412–5424, 2016. http://doi.org/10.1109/TGRS.2016.2564501.

[22] Limberger, F.A.; Oliveira, M.M.: Real-time detection of planar regions in unorganized point clouds. Pattern Recognition, 48(6), 2043–2053, 2015. http://doi.org/10.5753/ctd.2015.10000.

[23] Lu, Y.; Rasmussen, C.: Simplified markov random fields for efficient semantic labeling of 3d point clouds. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2690–2697. IEEE, 2012.

[24] Maturana, D.; Scherer, S.A.: Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg,

Germany, September 28 - October 2, 2015, 922–928. IEEE, 2015. http://doi.org/10.1109/IROS.2015.7353481.

[25] Morsdorf, F.; Meier, E.; Kotz, B.; Itten, K.I.; Dobbertin, M.; Allgower, B.: Lidar-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. Remote Sensing of Environment, 92(3), 353–362, 2004. http://doi.org/https://doi.org/10.1016/j.rse.2004.05.013.

[26] Nurunnabi, A.; Belton, D.; West, G.: Robust segmentation in laser scanning 3d point cloud data. In 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA), 18. IEEE, 2012.

[27] Park, J.C.; Shin, H.; Choi, B.K.: Elliptic gabriel graph for finding neighbors in a point set and its application to normal vector estimation. Comput. Aided Des., 38(6), 619–626, 2006. http://doi.org/10.1016/j.cad.2006.02.008.

[28] Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. CoRR, abs/1612.00593, 2016. http://arxiv.org/abs/1612.00593.

[29] Rusu, R.B.: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. Ph.D. thesis, Technische Universitat Munchen, 2010. https://mediatum.ub.tum.de/doc/800632/941254.pdf.

[30] Sampath, A.; Shan, J.: Clustering based planar roof extraction from lidar data. In American Society for Photogrammetry and Remote Sensing Annual Conference, 1–6, 2006.

[31] Sampath, A.; Shan, J.: Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. IEEE Transactions on Geoscience and Remote Sensing, 48(3), 1554–1567, 2010. http://doi.org/10.1109/tgrs.2009.2030180.

[32] Shapovalov, R.; Velizhev, E.; Barinova, O.: Nonassociative markov networks for 3d point cloud classification. In International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 38, 103–108, 2010.

[33] Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E.G.: Multi-view convolutional neural networks for 3d shape recognition. CoRR, abs/1505.00880, 2015. http://arxiv.org/abs/1505.00880.

[34] Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S.: Segcloud: Semantic segmentation of 3d point clouds. In 2017 International Conference on 3D Vision (3DV), 537–547. IEEE, 2017. http://doi.org/10.1109/3dv.2017.00067.

[35] Tenenbaum, J.B.; de Silva, V.; Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science, 290, 2319–2323, 2000. ISSN 0036-8075. http://doi.org/10.1126/science.290.5500.2319.

[36] The CGAL Project: Computational geometry algorithms library, 2022. https://www.cgal.org/.

[37] Van Kaick, O.: Wcseg: Shape segmentation by approximate convexity analysis. https://people.scs.carleton.ca/~olivervankaick/wcseg/, 2023. Accessed on May 12, 2023.

[38] Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M.: Octree-based region growing for point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing, 104, 88–100, 2015.

[39] Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph., 38(5), 146:1–146:12, 2019. http://doi.org/10.1145/3326362.

[40] Xie, Y.; Tian, J.; Zhu, X.X.: Linking points with labels in 3d: A review of point cloud semantic segmentation. IEEE Geoscience and Remote Sensing Magazine, 8(4), 38–59, 2020. http://doi.org/10.1109/MGRS.2019.2937630.

[41] Xu, L.Q.; Oja, E.; Kultanen, P.: A new curve detection method: randomized hough transform (rht). Pattern Recognition Letters, 11(5), 331–338, 1990. http://doi.org/10.1016/0167-8655(90)90042-z.

[42] Zhang, J.; Lin, X.; Ning, X.: Svm-based classification of segmented airborne lidar point clouds in urban areas. Remote Sensing, 5(8), 3749–3775, 2013. http://doi.org/10.3390/rs5083749.