Journal for Geometry and Graphics
Volume 30 (2026), No. 1, 97–116

Geometric Clustering of Point Clouds Using the Spheres Method in Parametric Design

Larysa Ivanova[1], Nataliya Sadretdinova[2], Zlata Tošić[1]

[1] *Dresden University of Technology, Dresden, Germany*
Kyiv National University of Construction and Architecture, Kyiv, Ukraine
larysa.ivanova@mailbox.tu-dresden.de, ivanova.ls@knuba.edu.ua,
zlata.tosic@tu-dresden.de

[2] *Dresden University of Technology, Dresden, Germany*
Kyiv State University of Technologies and Design, Kyiv, Ukraine
nataliya.sadretdinova@tu-dresden.de

Abstract. This article explores the use of non-hierarchical clustering methods for processing point clouds data in applied design and production tasks. The paper offers a detailed classification of clustering algorithms by their formation principles (e.g., partitioning, density-based, model-based, etc.) and introduces a geometric classification of point clouds based on their spatial distribution, dimensionality, and shape complexity.

The proposed clustering approach, referred to as the Spheres Method, is designed for the geometric interpretation of both initially spatial datasets (e.g., LiDAR, photogrammetry) and statistical or technological datasets transformed into point clouds through parameterization. The Spheres Method is particularly effective for identifying implicit groupings within uniform yet structurally diverse 2D/3D datasets. It enhances automation in point clouds segmentation and provides a scalable solution for optimizing design and manufacturing workflows.

The research focuses on two distinct cases: clustering 3D point clouds in the design of rehabilitation textile products and in the construction of a prefabricated garden pavilion shell using additive manufacturing.

Key Words: point clouds clustering, non-hierarchical clustering, spheres Method, parametric design, classification of clustering methods, point clouds geometry

MSC 2020: 62H30 (primary), 68U05, 65D18, 62R40

1 Introduction

The data generated during the operation of machines, mechanisms, processes, or in the creation of architectural and design objects require proper mathematical processing to identify

ISSN 1433-8157/

relationships between deviations, influencing factors, and resulting characteristics. A suitable tool for processing these data is cluster analysis. Automation of point clouds data processing is crucial for building efficient decision-making systems and reducing manual effort. Clustering is used in diverse fields including marketing, medicine, architecture, design, computer graphics, and the optimization of production processes.

Clustering refers to grouping data based on specific criteria such as type, size, shape, category, etc. The algorithm seeks a solution where all objects within a cluster are as similar as possible, and the clusters are as distinct as possible. Similarity is determined by a set of attributes or parameters – the clustering Criteria Vector. The purpose of clustering is to reveal the structure of data by segmenting a set of objects (points in the feature space) into homogeneous groups. This partitioning simplifies the analysis and allows applying the most appropriate processing methods to each cluster. Although hundreds of clustering algorithms exist, optimal solutions are only feasible for small datasets. As the number of data points grows, the task quickly becomes computationally challenging. According to Kleinberg's impossibility theorem [17], there is no universally optimal clustering algorithm, due to the wide variety of application contexts. Clusters vary in shape, density, and size, and data attributes can have different scales, units, and continuity. Therefore, clustering should be considered a research tool, which must be selected and fine-tuned depending on the problem.

Cluster analysis can be classified into two types: hierarchical and non-hierarchical [16]. Hierarchical clustering builds nested structures of clusters, while non-hierarchical clustering divides the data into a fixed number of groups, assigning each data point to exactly one cluster. In non-hierarchical clustering, the relationship between clusters is not defined. Non-hierarchical clustering is an iterative process that partitions a dataset into a predefined number of clusters. Algorithms differ in their choice of initialization, cluster formation rules, and stopping criteria. Two main approaches exist: identifying dense regions in the data space or minimizing object dissimilarity.

This paper focuses on non-hierarchical clustering, applied to two case studies: (see Section 3) clustering of textile design data for rehabilitation purposes and (see Section 4) clustering in the formation of a pavilion shell structure composed of additive-manufactured components. It should be noted that the proposed in this paper Spheres method has potential for broader applications (see Section 2).

Non-hierarchical clustering encompasses a broad range of algorithmic approaches:

- Partitioning methods such as k-means [4, 14] and k-medoids [4, 21] iteratively optimize cluster assignments to minimize intra-cluster variance and maximize inter-cluster separation.
- Density-based methods such as DBSCAN [6, 7, 9, 12] and SUBCLU [15] identify clusters as contiguous regions of high point density, effectively handling noise and arbitrary cluster shapes.
- Grid-based methods such as STING and CLIQUE [2] discretize the data space into cells and assess point density within each cell.
- Model-based methods such as Gaussian Mixture Models [26] and the Expectation-Maximization (EM) algorithm [8] fit probabilistic models to the data distribution.
- Mode-seeking methods such as Mean Shift [13] locate clusters by iteratively shifting points toward local maxima in the estimated density function.
- Graph-based methods such as Spectral clustering [3] exploit the eigen structure of similarity graphs to partition datasets into communities.
- Geometric approaches such as Voronoi-based methods [27] define cluster boundaries

through spatial tessellation.
- Subspace-based methods such as CLIQUE [2] detect clusters in lower-dimensional projections of the data.
- Consensus-based methods such as CSPA and Monti [20] integrate clustering results obtained from multiple algorithms or repeated runs on the same dataset.

Modern visualization and pattern recognition tools often complement these methods. By combining spatial and temporal feature analysis with statistical and visual techniques, they enable more effective interpretation of clustering results and facilitate the detection of latent patterns in multidimensional datasets.

In architectural, design, and other geospatial applications, two types of input data are commonly encountered:

1. Geometric point clouds obtained through 3D scanning, photogrammetry, or LiDAR, where each point directly represents a real-world object in space.
2. Statistical or production-related data that must be converted into a geometric interpretation, where each point models a structural element described by a set of parameters (e.g., value, size, time, category) encoded as coordinates in a parametric space.

In both cases, the clustering task can be interpreted geometrically, allowing the use of universal spatial grouping methods regardless of data origin. In clustering tasks, a point clouds represents a set of spatial points in Euclidean space, characterized by specific geometric and topological properties. These properties significantly influence the selection and performance of clustering algorithms.

Table 1 presents a classification of point clouds according to several key attributes: dimensionality, density, topology, connectivity, and local structure. Such systematization helps to choose the most efficient clustering technique depending on the data type: from 2D projections to complex 3D or high-dimensional scattered structures.

Real-world point clouds often exhibit a combination of these characteristics. For instance, a point clouds can be non-uniform + branching + noisy, and require hybrid-clustering approaches.

The necessity for accurate clustering in various industrial and research applications continues to stimulate the development of original approaches that are specifically adapted to the characteristics of individual cases. This observation supports the author's concept regarding the impossibility of a universal clustering method, which, in turn, explains the successful outcomes of the experiments conducted in this study. For instance, the approach proposed in [19] concerning the formulation of a clustering function for neural network training and the segmentation of trees within forest environments described in [10] cannot be directly compared, at least due to the differences in the nature of the input data and the expected objectives of each task. Of particular note is the approach presented in [18] and [11], which emphasizes the preliminary simplification of point clouds prior to clustering in the context of cultural heritage analysis. This method combines clustering with the assessment of the significance of the input data, which is critically important for the accurate reconstruction of object geometry. It is evident that the high noise level and structural complexity of scanned point clouds in such studies necessitate preliminary data filtering, where researchers must find a balance between preserving informative points and removing excessive noise. The study described in [1], which focuses on ensuring reliable system operation under dynamic transportation conditions, rightly highlights the necessity of a systematic review and classification of existing clustering methods. This need arises from the specific challenges inherent in LiDAR data, such as high point density, noise, and variable sampling frequency.

Table 1: Point cloud types and recommended clustering methods

No.	Criterion	Type of Point Clouds	Description / Examples	Suitable Clustering Methods
1	Dimensionality	1D (embedded in a curve)	Lines, trajectories, trend paths	Spectral clustering, linear models
		2D	Surfaces, floor plans, projections	K-means, DBSCAN, region growing
		3D	Volumetric forms, 3D scanned objects	DBSCAN, Octree-based, HDBSCAN
		High-dimensional (projected to 2D/3D)	Feature-rich or parametric data (e.g., t-SNE, UMAP projections)	Gaussian Mixture, Spectral clustering, K-means, Spheres Method
2	Density	Uniform	Evenly distributed, grid-like	K-means, grid-based, Spheres Method
		Non-uniform	Varying density across regions	HDBSCAN, adaptive DBSCAN
		With noise and outliers	Isolated, random noise points	Robust clustering, DBSCAN
3	Shape / Topology	Isolated compact clusters	Well-separated groups	K-means, DBSCAN
		Elongated / stripe-like	Line-like or banded shapes	Spectral, agglomerative, skeleton-based, Spheres Method
		Circular / closed-loop	Loops, contours	Spectral clustering, Isomap
		Branching structures	Trees, branching paths	Graph-based, region growing, trajectory-based
4	Connectivity	Connected point set	All points are reachable by a continuous path	Graph-based, flood fill, region growing, Spheres Method
		Fragmented or disjoint	Multiple disconnected components	DBSCAN, hierarchical
5	Local Structure	Smooth forms	Gently curved or flat surfaces	Gaussian Mixture, RBF-based models, Spheres Method
		Angular / sharp-edged	Edges, corners, abrupt changes	Region growing, RANSAC
		Fractal / highly detailed	Irregular or chaotic, fine structure	Spectral, HDBSCAN

Thus, contemporary research in the field of clustering underscores the importance of a preliminary analysis of point cloud characteristics and task objectives prior to selecting a specific clustering method. It cannot be said that a method is better or worse out of context – everything is determined by the task, the characteristics of the data and the purpose of the analysis. This is especially true in the context of applied design, architecture and digital fabrication, where the geometric nature of the data and the task context play a key role.

The Spheres method proposed in this paper is suited for 2D/3D spatially-extended point clouds with relatively uniform distribution and implicitly defined point groups.

2 The Spheres Method: A geometric clustering approach for 2D/3D point clouds

This paper introduces a novel clustering approach – the Spheres method, grounded in a geometric interpretation of the segmentation process. Its main advantages are implementation simplicity and computational efficiency compared to traditional algorithms.

The Spheres Method is a non-hierarchical, iterative clustering algorithm based on minimizing object dissimilarity. It is scalable, robust to noise, and effective with varying cluster densities, as membership is determined solely by the radius of a cluster-forming sphere, while discontinuities between points are mitigated by selecting centroids from an ordered list. This algorithm has been implemented in Rhino and Grasshopper, one of the most widely used platforms for parametric modeling. This environment provides efficient and accurate tools for the calculation and visualization of complex geometric forms, thereby facilitating modeling and analysis in practical applications.

Input Data Characteristics and Problem Statement The choice of the method is due to the nature of the input data – point clouds obtained by transforming the Criteria vector of product specifications into coordinates of the parametric space.

Product specifications determined based on expert assessments (see Sections 3 and 4) are represented as a cloud of two- or three-dimensional points. Each point models an individual object or element within the system. The resulting point clouds exhibits uniform density without visually apparent segmentation tendencies and typically forms elongated bands (Figures 4 and 9).

Due to varying dimensionalities and scales of the input parameters, all values normalized to the $[0, 1]$ range. Normalization eliminates disproportionate parameter influence, accelerates convergence, and improves prediction accuracy.

Neither the number of clusters nor the object count per cluster is known a priori, necessitating their determination as part of the problem formulation.

Choice of Clustering Method In practice, the choice of a clustering algorithm requires testing several approaches and comparing their outcomes. One of the reliable strategies for validating a new method is to compare it with well-established algorithms.

Considering the characteristics of the input data and the classification of clustering methods presented in Section 1, two widely used algorithms, k-means and DBSCAN, were selected for comparison with the proposed Spheres Method. The k-means algorithm is well suited for datasets with uniform density and clusters of nearly spherical shape, whereas DBSCAN is more effective for structures with non-uniform density and clusters of arbitrary form.

In our experiments, k-means, which requires prior specification of the number of clusters, exhibited low stability when applied to uniformly dense elongated point clouds. Different centroid initializations and values of k led to inconsistent results: points near the cluster boundaries frequently shifted between groups due to the absence of visually distinct separations. Nevertheless, with careful parameter tuning, k-means was able to produce results fully consistent with the partitioning obtained by the Spheres Method. This outcome simultaneously confirms the validity of the proposed method and highlights a key drawback of k-means: its instability across multiple runs.

The Spheres Method, by contrast, provides stable results. However, for clusters with particularly diffuse boundaries, it may generate several equally plausible grouping alternatives. The selection of the optimal solution can then be guided by the silhouette coefficient in combination with domain-specific requirements and production-related economic priorities. This variability constitutes an advantage of the Spheres Method, as it offers experts a range of viable clustering scenarios to support informed decision-making.

DBSCAN, although capable of identifying clusters of arbitrary shape and density, requires careful parameter tuning and performs poorly on uniformly distributed data (see Figures 5 and 10). In the conducted experiments, domain experts negatively evaluated the clustering results obtained with DBSCAN (see Sections 3 and 4).

Thus, the comparative analysis confirms the accuracy, stability, and adaptability of the Spheres Method, establishing it as a more reliable tool for clustering uniformly dense elongated spatial structures.

Principle of the Spheres Method The Spheres Method was developed with a focus on the geometric properties of the point cloud. Clustering is based on estimating the "distance" between points – the degree of similarity. The Euclidean distance metric is used for the method of spheres.

The input dataset is a finite set of points:

$$X = \{x_1, x_2, \ldots, x_n\}, \quad x_i \in \mathbb{R}^d,$$

where $d = 2$ or $d = 3$ denotes the dimensionality of the feature space.

In the first iteration, all points $x_i \in X$ are sorted along a selected coordinate axis (e.g., the x-axis). The first point in the sorted list becomes the *centroid* of the first cluster C_1:

$$c_1 = x_{i_1}.$$

A hypersphere of radius r is constructed around the centroid c_1, and the first cluster is defined as:

$$C_1 = \{x \in X \mid \|x - c_1\| \leq r\}.$$

All clustered points are then removed from the dataset:

$$X' = X \setminus \bigcup_{j=1}^{k} C_j.$$

The procedure is repeated for the next first point in the remaining set X', which is not yet assigned to any existing cluster. This process continues until the remaining set is empty, i.e., $X' = \varnothing$.

The radius r simultaneously determines both the number of clusters k and the number of points in each cluster. By varying r, multiple cluster configurations $\{C_1, C_2, \ldots, C_k\}$ can be generated and compared using internal validation metrics.

In the second stage, the cluster structure is refined. For each cluster C_j, the centroid is recomputed as the mean of all its points:

$$c_j = \frac{1}{|C_j|} \sum_{x \in C_j} x,$$

and the radius is updated as the distance from the centroid to the farthest point in the cluster:

$$r_j = \max_{x \in C_j} \|x - c_j\|.$$

This two-step scheme (Figure 1) enables adjustment of borderline or ambiguous points and improves the method's robustness to noise and variations in local density. The output data is a three-dimensional graph where data points belonging to different clusters are represented by different colors (Figures 4 and 9).

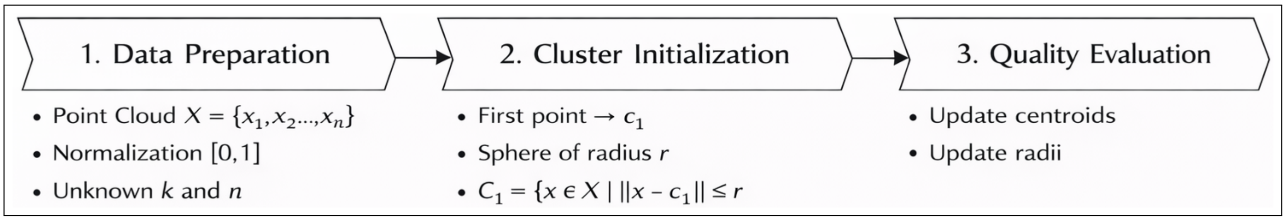

Figure 1: Stages of data clustering using the Spheres method.

The method demonstrates fast convergence, stable results, and low computational complexity, even on large datasets. Its robustness arises from the geometric nature of the clustering procedure and the absence of a need for a predefined number of clusters.

Clustering Quality Assessment Assessing clustering quality is essential to determine the optimal number of clusters and to evaluate how well the segmentation reflects the underlying data structure. In the absence of prior knowledge about cluster structure (as in the current case), internal validation metrics are used, based solely on the internal relationships within the data. To estimate the optimal number of clusters, the following techniques were employed:

- The Elbow Method [25], based on plotting the within-cluster sum of squares (WCSS) against the number of clusters k;
- The Silhouette Coefficient Method [22], which quantifies the average proximity of each point to its own cluster versus the nearest alternative.

Within-Cluster Sum of Squares (WCSS) measures cluster compactness and decreases as the number of clusters increases:

$$\text{WCSS} = \sum_{j=1}^{k} \sum_{i=1}^{n} \min\left(\|x_i^{(j)} - c_j\|\right)^2,$$

where:

- k – number of clusters,
- n – number of observations,
- $x_i^{(j)}$ – the i-th observation in the j-th cluster,
- c_j – the centroid of cluster j.

The optimal number of clusters is found at the "elbow point" in the WCSS curve, where further increases in k yield diminishing improvements. To eliminate the subjectivity of the WCSS graph interpretation, it is proposed to use the curvature diagram of the curve approximating the WCSS values. Extreme values of the diagram allow us to accurately determine the optimal value of k (Figures 3 and 8).

The Silhouette Coefficient integrates both cohesion (how closely related points are within a cluster) and separation (how far they are from points in other clusters). The average silhouette score across all cluster configurations serves as an additional metric for determining the optimal number of clusters. Unlike the smooth WCSS curve, the silhouette score graph typically has a distinct peak. Both metrics provided consistent results: the elbow point of the WCSS curve and the maximum silhouette score corresponded to the same optimal number of clusters (Figures 3 and 8).

The quality of point segmentation within a cluster was controlled by expert judgment using Silhouette (The silhouette coefficient values). The silhouette coefficient for each sample i is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i),\, b(i))},$$

where:

- $a(i)$ – the average distance from i to all other points in the same cluster (intra-cluster distance),
- $b(i)$ – the lowest average distance from i to points in any other cluster (inter-cluster distance).

Silhouette values range from -1 to 1. Values close to 1 indicate well-clustered points; values around 0 suggest overlapping clusters; and negative values indicate potential misclassification. Average silhouette values in the [0.4–0.7] range indicate acceptable cluster structure for practical use (Figures 4 and 9).

Despite the existence of many algorithms, clustering results may differ significantly. Therefore, expert evaluation remains the most important tool for verifying the quality of clustering. In Sections 3 and 4 we present examples of practical problems in which the Spheres Method has demonstrated high efficiency, stability of the result, and also allowed to achieve significant optimization of production. This is confirmed by both numerical quality metrics and expert evaluations (Table 2).

Table 2: Comparison of clustering results in two experimental case studies

Criterion / Experiment	**Spheres Method**	**k-means Method**	**DBSCAN Method**
Experiment 1 — Clustering of Belt Patterns			
Input data	2D point cloud of 27 standardized belt patterns	Same data	Same data
Optimal number of clusters (k)	4 or 6 (confirmed by the Elbow and Silhouette methods)	Sensitive to initialization; unstable results	Irregular clusters; poor adaptation to uniform density
Average Silhouette coefficient	0.4–0.7 (acceptable quality)	Similar values, but unstable	Low or negative values
Expert evaluation	Positive; stable and interpretable groups	Acceptable, but inconsistent	Negative; unsatisfactory cluster shapes

Continued on next page

Table 2 – continued from previous page

Criterion / Experiment	**Spheres Method**	**k-means Method**	**DBSCAN Method**
Practical outcome	Six-cluster configuration selected – stable size categorization for production	Requires manual tuning of k	Clusters unsuitable for standardized sizing
Experiment 2 — Clustering of Modular Shell Elements			
Input data	3D geometric feature vectors (radius, difference of angles, difference of side lengths)	Same data	Same data
Optimal number of clusters (k)	4 (confirmed by the Elbow method; Silhouette coefficient = 0.675)	Similar result, but less distinct boundaries	Two small clusters; unstructured results
Cluster shape	Smooth geometric transition from center to periphery	Rough separation	Disconnected, irregular shapes
Expert evaluation	Fully consistent with geometric logic	Moderate consistency	Unsatisfactory (negative silhouette values)
Practical outcome	Four regularized module types; reduced formwork cost	Acceptable but less efficient	Unsuitable for production optimization
Overall assessment	High stability, geometric interpretability, scalable for design applications	Parameter-dependent; unstable for elongated data	Poor performance on uniformly distributed data

3 Clustering of Belt Pattern (BP Project)

The objective of this project is to develop and test a methodology for designing functionally specific textile products intended for mass production and close fitting to the lower part of the human body. As input data, parametric human models of standard sizes (from 38 to 54) were used, with three hip-width variations – narrow, normal, and wide. Based on these models, 2D flat garment patterns were created using 3D design tools (Figure 2).

A total of 27 size patterns were obtained and standardized, taking into account the geometric features of bodies of various shapes. However, a large number of distinct sizes introduces an additional level of complexity in mass production within the framework of established size standardization systems. For form-fitting garments, precise alignment between the product geometry and the body surface is critical. Potential mismatches may be partially compensated by the elasticity of materials or adjustable structural elements. This makes clustering a relevant task, namely, optimally grouping the set of patterns into clusters with a high degree of similarity. This allows for the generation of averaged templates within each group and the definition of standard size categories that correspond to typical consumer segments. Such an approach strikes a balance between production efficiency and the precision

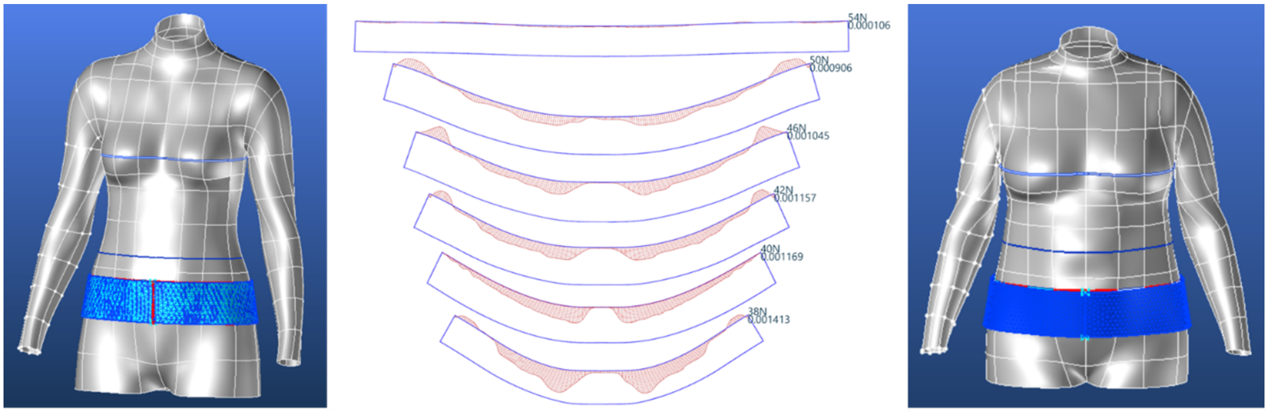

Figure 2: Generation, unfolding, and curvature analysis of belt patterns in various sizes.

of garment fit.

A key step involved formalizing a Criteria vector to describe pattern similarity. One of the clustering criteria was the curvature of the boundary lines forming the contour of the pattern. An additional criterion was the length of the pattern, with a maximum allowed deviation of 6 cm within a cluster. All parameter values were normalized using min-max scaling on the range $[0, 1]$ to eliminate scale-related distortions and unit inconsistencies. As a result, the data formed a band of evenly distributed points in a two-dimensional space. According to expert estimation, the expected number of clusters was in the range of 4 to 7, with one of the conditions being a relatively balanced number of patterns per cluster.

The approach we used to determine the optimal values for regular sizes in belt manufacturing was presented at the 3DBodytech Conference in Lugano in 2025, where we introduced a clustering strategy for analysing and grouping belt pattern outlines [23].

In the follow-up work, the earlier methodology was revised and implemented to optimise the size range used in manufacturing. While the core elements, such as the parametric human model framework and the 3D belt design process, were preserved, the clustering stage was reworked to align more closely with production demands.

The Sphere Method, we applied in the current stage, achieves a higher degree of fit precision for a wide spectrum of body types, without compromising the efficiency or scalability required for mass production. The approach identified two stable configurations: clustering into four and into six groups. The results are visualized in a scatter plot where each data point is colored according to its cluster membership. For clarity, the cluster-forming spheres from the method's second iteration are also shown on the plot. The selection of the optimal number of clusters was supported by internal validation metrics: Elbow method and Silhouette Coefficient method. Although the WCSS curve is relatively smooth, the curvature diagram of the approximating function revealed two distinct extrema at four and six clusters. The Silhouette score plot also indicated peaks for the same values (Figure 3). However, considering production-related factors, expert evaluation favored the 6-cluster configuration (Figure 4 bottom). In this case, each cluster adequately represents a specific size group: the length deviation does not exceed 6 cm, and curvature differences remain within one percent. This configuration ensures complete coverage of body shapes within the defined size range and guarantees that the resulting pattern templates will accurately conform to the anatomical geometry of the lower supporting surface of the body.

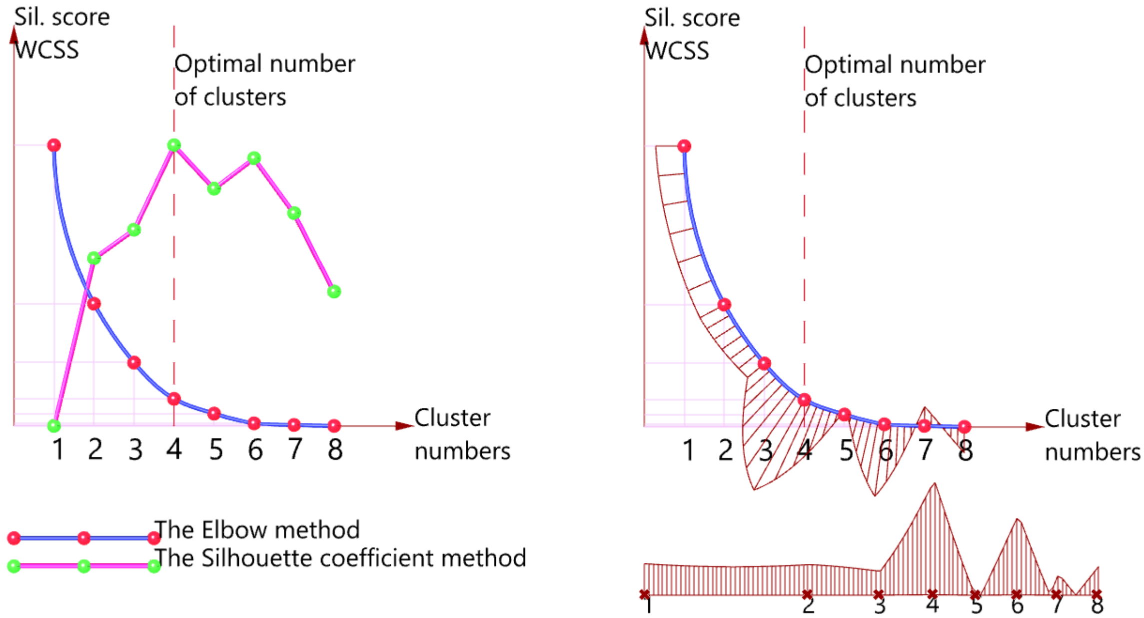

Figure 3: BP project clustering results: Finding the optimal number of clusters by Elbow method and Silhouette Coefficient method.

Silhouette analysis for both configurations yielded average coefficients in the range of [0.4–0.7], which corresponds to acceptable clustering quality. The 4-cluster solution demonstrated slightly better segmentation accuracy at the level of individual points (Figure 4 top).

To compare, clustering using DBSCAN algorithm, applied in the subsequent study, allowed identifying groups of densely located points, but for the purposes of this task the obtained clusters were considered less suitable due to their irregular shape and uneven distribution of points within the clusters (Figure 5). The expert evaluation was negative.

Owing to the applied relevance of the findings, the proposed methodology demonstrates a high potential for direct implementation within the apparel manufacturing sector. The integration of the Sphere Method into the established design workflow has resulted in a pattern development process that combines a high degree of technical accuracy with improved adaptability to a broad spectrum of anthropometric profiles. This is particularly advantageous for applications that require a stable and uniform interface between the product and the body surface. The authors successfully applied the developed approach to create garments based on belts adapted to the geometry of the female body, including the Adaptable Harness-Based Carrying System for Patients with Dementia Symptoms [24] and a Lightweight Load-Carrying System for Female Military Personnel.

The applicability of the present approach to garment design offers significant opportunities for advancing the precision of analytical procedures involving the classification and grouping of multiple interdependent parameters. Within industrial sizing systems, anthropometric research commonly employs primary body descriptors such as body mass, stature, and key circumferential and linear measurements. The objective of such analyses is to achieve an optimal distribution of these parameters across standardised size categories, thereby ensuring both functional fit and population coverage [5].

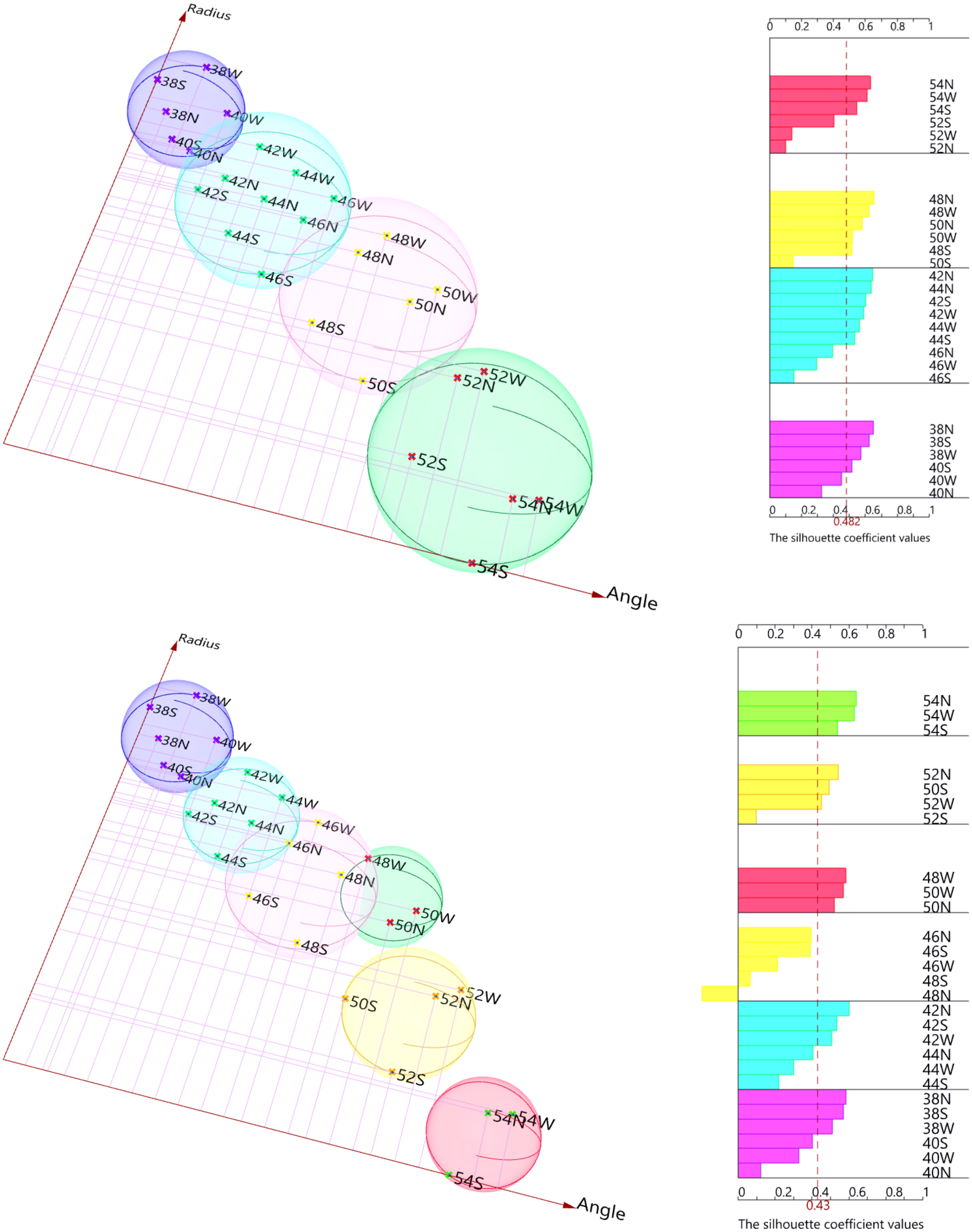

Figure 4: BP project clustering results: top – The 4-cluster solution and Silhouette graph using Spheres method; bottom – The 6-cluster solution and Silhouette graph using Spheres method.

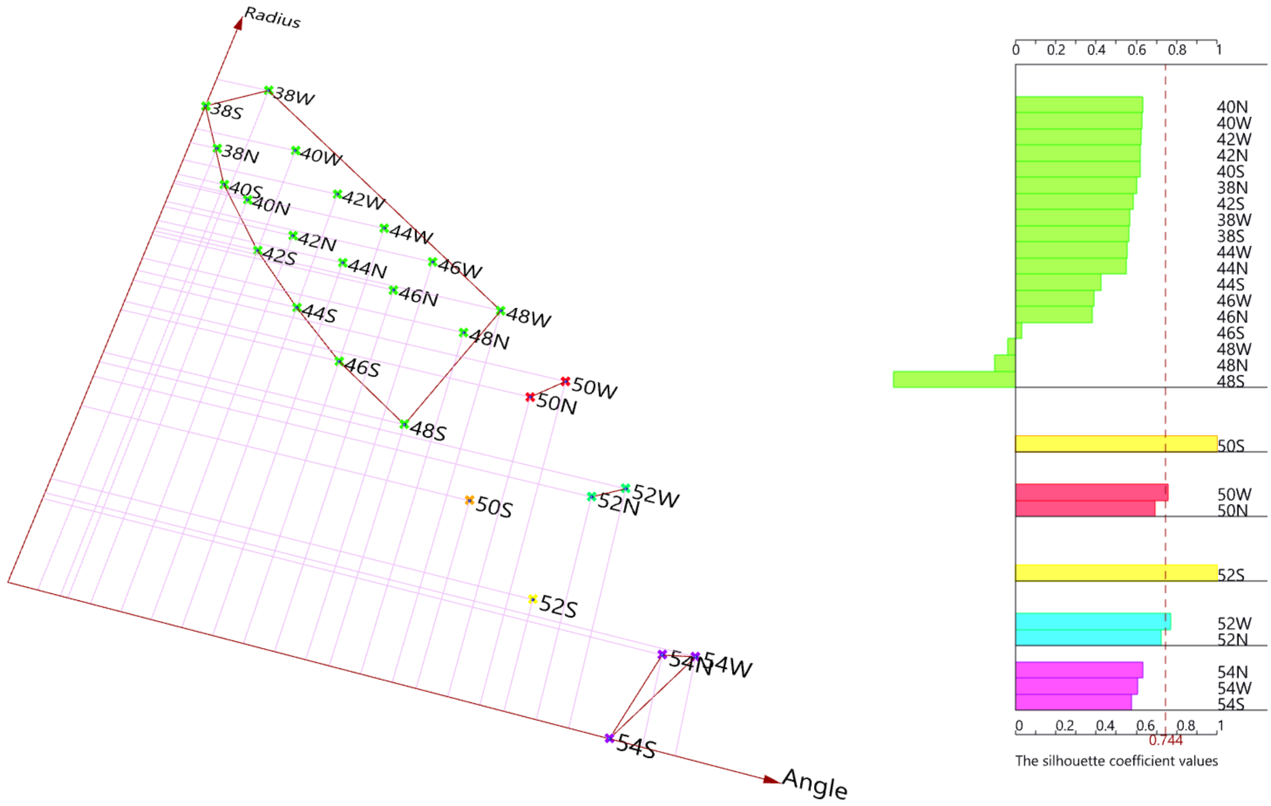

Figure 5: BP project clustering results: The 6-cluster solution and Silhouette graph using DBSCAN method.

4 Clustering of Modular Shell Elements for a Garden Pergola (MSEforGP Project)

The project aims to design and fabricate a modular shell structure for a garden pergola, composed of hexagonal reinforced concrete blocks produced using formwork. The shell consists of 29 blocks, which were initially classified by an expert into 11 distinct types based on their size and shape differences. The geometry of each block was derived from a regular hexagonal grid and then projected onto a sphere using inverse stereographic projection, which generates the curvature of the structure. Because the grid has three-axis symmetry only in the plane, only the central module retains the shape of a regular hexagon. All other modules deviate from regularity due to variations in corner angles and side lengths, adapting to the curvature of the surface (Figure 6).

The objective of the task was to optimize production by reducing the number of unique block types, thereby decreasing manufacturing time and costs of the formwork. To achieve this, clustering was performed using the Spheres method, which groups similar elements based on geometric similarity.

The objects of analysis were volumetric hexagonal modules. Each was described by a three-dimensional Criteria vector comprising:

- the radius of the circumscribed circle;
- the difference between the maximum and minimum interior angles;
- the difference between the maximum and minimum side lengths.

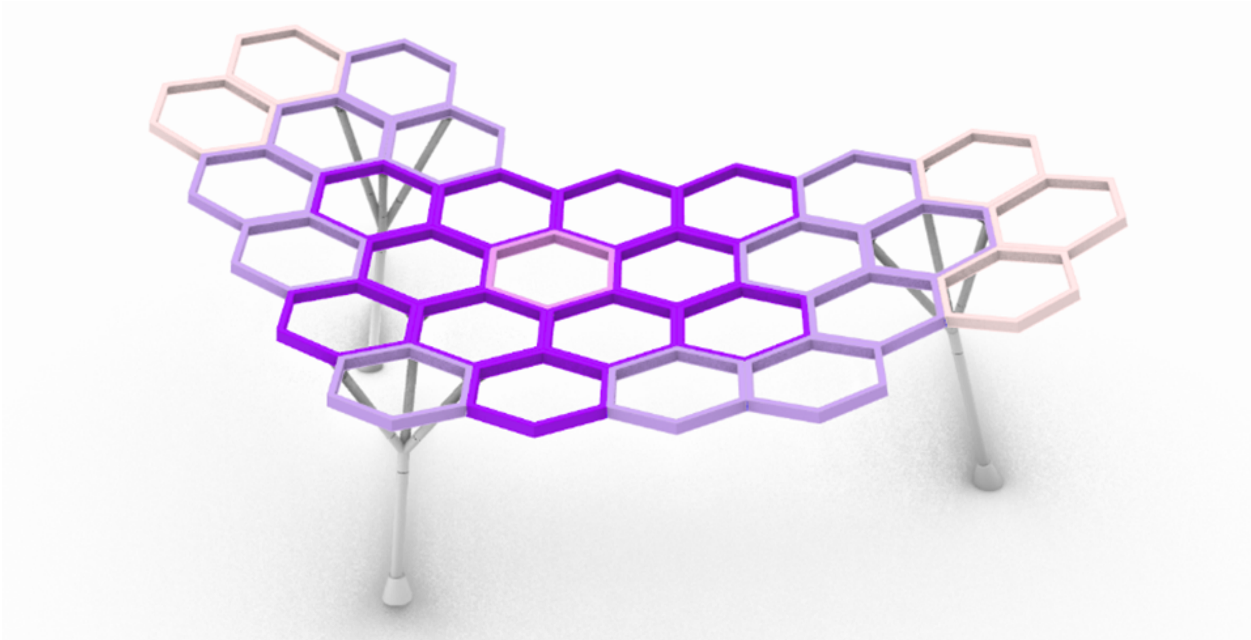

Figure 6: Concrete modular shell with clustered hexagon modules.

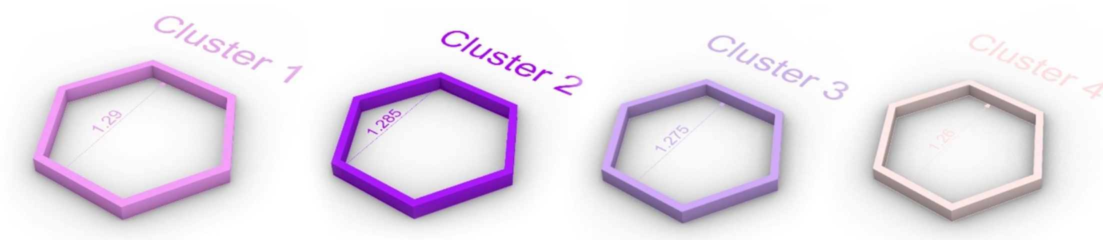

Figure 7: MSEforGP project clustering results: The given sizes of the hexagon modules for four clusters.

Since the selected features had different units and scales, the data were normalized using min-max scaling to the range $[0, 1]$. This allowed the input to be represented as a 3D point cloud in feature space, where each point corresponded to a specific block model. In this case, no prior information about the number of clusters, the location of centroids, or the distribution of models was available. Visual inspection revealed no obvious grouping. The expert acknowledged that clusters might vary in size and did not need to be uniformly populated. Analysis of the WCSS graph revealed an elbow at $k = 4$, indicating the presence of four optimal clusters. This finding was supported by a local extremum in the curvature diagram of the approximated WCSS curve, as well as peak values in the Silhouette Score metric (Figure 8).

The quality of the segmentation within clusters was further assessed using the silhouette coefficient. The results were well balanced: most coefficient values ranged from 0.4 to 0.7, indicating satisfactory group separation and aligning with visual observations (Figure 9).

The first cluster included only the regular hexagon at the center of the shell – a logical outcome given its function as the geometric core of the structure (Figures 6 and 7). The remaining blocks were grouped according to their degree of deviation from regularity, forming a gradual radial transition from the center toward the shell's periphery. The final clustering

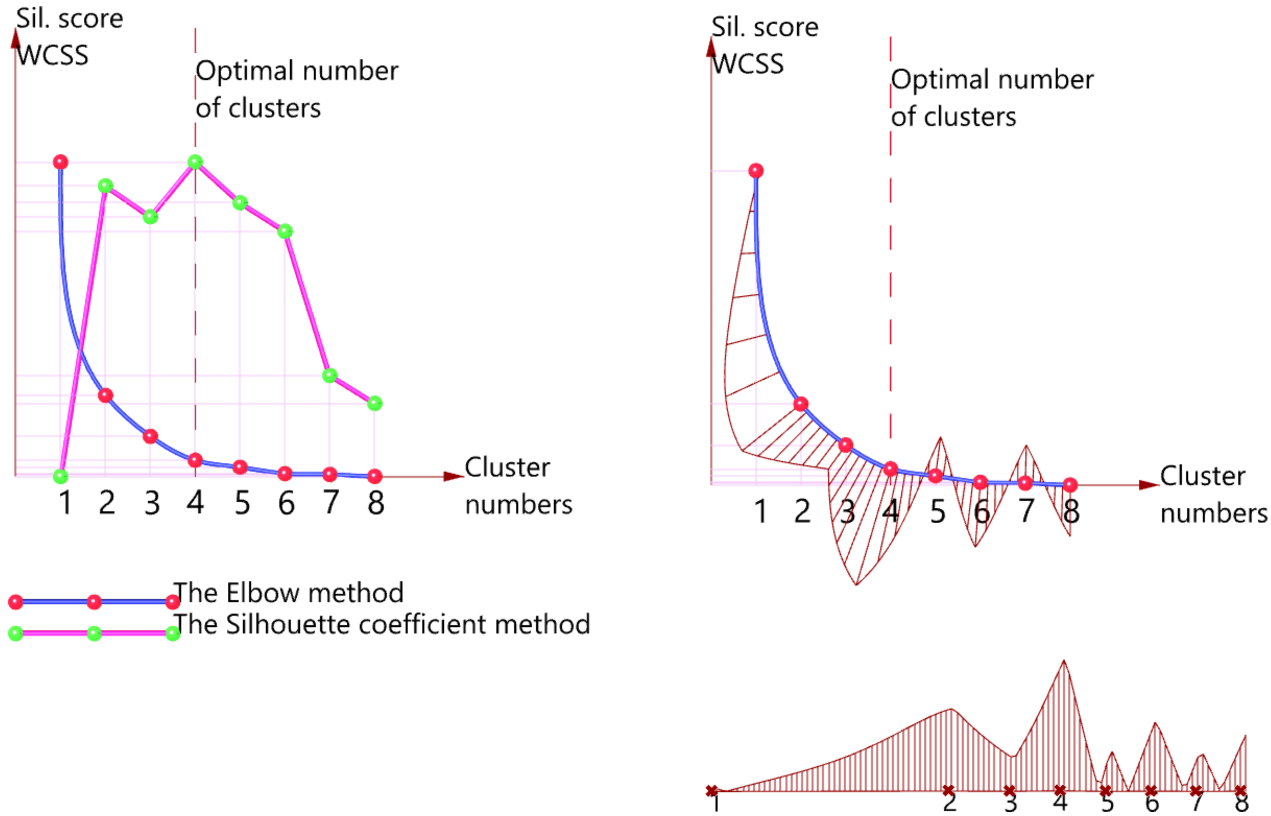

Figure 8: MSEforGP project clustering results: Finding the optimal number of clusters Elbow method and Silhouette Coefficient method.

outcome aligned with expert expectations. Moreover, averaging the geometry within each cluster improved the alignment of neighboring blocks, leading to smoother transitions and enhancing the overall structure of the shell.

The DBSCAN algorithm was also tested for comparison, but it gave unsatisfactory results: two clusters with only one element were identified, and the silhouette coefficients showed high variability with negative outliers (Figure 10).

Thus, the Spheres method demonstrated strong performance in clustering geometrically complex objects defined by a three-dimensional Criteria vector, particularly in cases where no clear grouping tendency is visually apparent.

It is also important to note that once the different modules were assigned to clusters, a single regular module size was defined for each cluster to further simplify the formwork geometry. This means that the slightly skewed modules were adjusted toward a regular shape, which in turn required different gap sizes between them in the curved, global shape. The diameters of the modules were varied within each cluster to determine the most suitable size.

The first priority was ensuring that the elements did not overlap, and the second was to minimize the gaps between modules. Additional experiments were carried out with a large number of clusters to verify the possibility of further reducing gaps. However, for this composition of elements, the maximum gap size did not decrease sufficiently to justify the additional formwork that would have been required. Based on these criteria, it was determined that the optimal number of clusters is four, with a silhouette coefficient of 0.675.

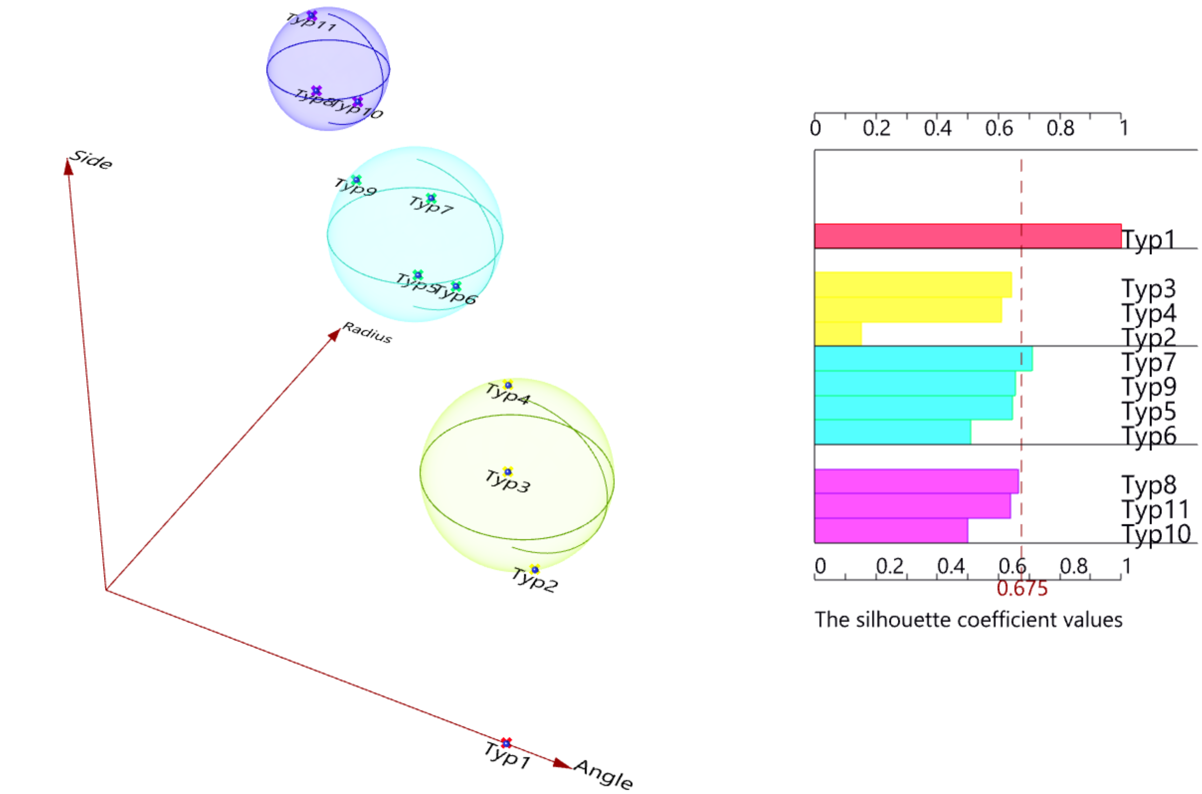

Figure 9: MSEforGP project clustering results: The 4-cluster solution and Silhouette graph using Spheres method.

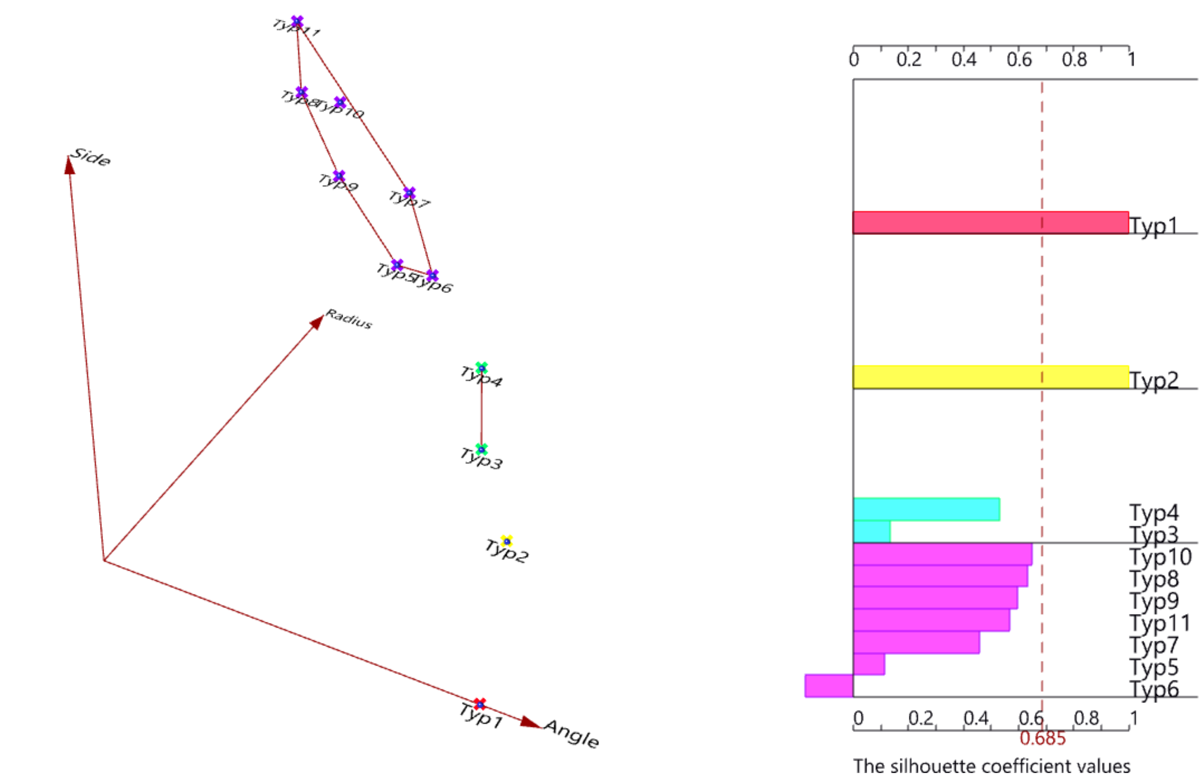

Figure 10: MSEforGP project clustering results: The 4-cluster solution and Silhouette graph using DBSCAN method.

5 Conclusions and Discussion

This study introduces a structured classification system for point clouds data, emphasizing their geometric and metric characteristics. This taxonomy enables a more precise and task-specific selection of clustering algorithms, tailored to the spatial complexity and form of the dataset. Through a comprehensive review of contemporary non-hierarchical clustering techniques, the paper highlights their relative advantages and limitations, particularly in the context of parametric modeling, digital fabrication, and computational geometry (Table 1, Section 1).

The key point of this paper is that no universal clustering method exists. This is supported by Kleinberg's impossibility theorem. Therefore, algorithm selection must be informed by the properties of the data: cluster shape, density, level of noise, and the structural goals of the analysis. Clustering should not be treated as a one-size-fits-all tool, but as a carefully tuned method depending on the data type and application context.

The article outlines two distinct classes of clustering problems. The first involves geometric input data, such as point clouds obtained from 3D scanning, photogrammetry, or LiDAR. The second deals with statistical or process-based datasets that must first be transformed into geometric representations. In both scenarios, a spatial reinterpretation of data proves essential for achieving meaningful and actionable clustering outcomes.

The proposed clustering method was applied in two real-world case studies to test its validity and effectiveness. One case focused on the design of textile rehabilitation products, while the other involved parametric modeling of a modular canopy structure for a garden pavilion manufactured using additive technologies. In both applications, expert evaluation confirmed the method's ability to deliver a logical and accurate segmentation of data. A comparison of the clustering results in two experimental tasks is presented in Table 2 (Section 2). These clustering results significantly improved production efficiency and reduced complexity, leading to measurable cost savings and better process control.

Overall, the proposed methodology not only improves decision-making in complex design and manufacturing workflows but also demonstrates potential for broader application across architecture, engineering, product design, and other data-intensive disciplines. The combination of geometric classification and targeted clustering strategies presented in this research offers a scalable foundation for future developments in adaptive computational methods.

Acknowledgment

The authors gratefully acknowledge to Prof. Dr.-Ing. Daniel Lordick for providing valuable suggestions that significantly improved this paper.

Authors' Contributions

Each author contributed significantly to the development of this research. The specific contributions are outlined below:

Larysa Ivanova: conceptualization of the study, development of the clustering method, writing of the main manuscript draft, software development in Grasshopper/Rhino and interpretation of experimental results.

Nataliya Sadretdinova: expert validation of the method in applied task, data collection and preprocessing, preparation of illustrations and diagrams, and contribution to case study application “Clustering of the belt pattern outlines”.

Zlata Tošić: data collection and preprocessing, expert validation of the method in applied design tasks, preparation of illustrations and diagrams, and contribution to case study application “Clustering for the Prefabricated Shell of a Garden Pergola”.

References

[1] M. Adnan, G. Slavic, D. Martin Gomez, L. Marcenaro, and C. Regazzoni: *Systematic and Comprehensive Review of Clustering and Multi-Target Tracking Techniques for LiDAR Point Clouds in Autonomous Driving Applications*. Sensors **23**, 2023. ISSN 1424-8220. doi: 10.3390/s23136119.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan: *Automatic subspace clustering of high dimensional data for data mining applications*. SIGMOD Rec. **27**(2), 94–105, 1998. ISSN 0163-5808. doi: 10.1145/276305.276314.

[3] E. Arias-Castro, G. Chen, and G. Lerman: *Spectral clustering based on local linear approximations*. Electronic Journal of Statistics **5**, 2011. ISSN 1935-7524. doi: 10.1214/11-ejs651. http://dx.doi.org/10.1214/11-EJS651.

[4] P. Arora, Deepali, and S. Varshney: *Analysis of K-Means and K-Medoids Algorithm For Big Data*. vol. 78, 507–512. 2016. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2016.02.095. 1st International Conference on Information Security & Privacy 2015.

[5] K. Bartol, D. Bojanić, T. Petković, and T. Pribanić: *A Review of Body Measurement Using 3D Scanning*. IEEE Access **9**, 67281–67301, 2021. doi: 10.1109/ACCESS.2021.3076595.

[6] D. Birant and A. Kut: *ST-DBSCAN: An algorithm for clustering spatial–temporal data*. Data & Knowledge Engineering **60**(1), 208–221, 2007. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2006.01.013. Intelligent Data Mining.

[7] H. Chen, M. Liang, W. Liu, W. Wang, and P. X. Liu: *An approach to boundary detection for 3D point clouds based on DBSCAN clustering*. Pattern Recognition **124**, 108431, 2022. ISSN 0031-3203. doi: 10.1016/j.patcog.2021.108431.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin: *Maximum Likelihood from Incomplete Data Via the EM Algorithm*. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22, 2018. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1977.tb01600.x.

[9] D. Deng: *DBSCAN Clustering Algorithm Based on Density*. In *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, 949–953. 2020. doi: 10.1109/IFEEA51475.2020.00199.

[10] W. Ding, R. Huang, W. Yao, W. Zhang, M. Heurich, and X. Tong: *A simple oriented search and clustering method for extracting individual forest trees from ALS point clouds*. Ecological Informatics **86**, 102978, 2025. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2024.102978.

[11] S. Emadi and M. Limongiello: *Optimizing 3D Point Cloud Reconstruction Through Integrating Deep Learning and Clustering Models*. Electronics **14**(2), 2025. ISSN 2079-9292. doi: 10.3390/electronics14020399.

[12] M. Ester, H. Kriegel, J. Sander, and X. Xu: *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In *Knowledge Discovery and Data Mining*. 1996. https://api.semanticscholar.org/CorpusID:355163.

[13] K. Fukunaga and L. Hostetler: *The estimation of the gradient of a density function, with applications in pattern recognition*. IEEE Transactions on Information Theory **21**(1), 32–40, 1975. doi: 10.1109/TIT.1975.1055330.

[14] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhija, and J. Heming: *K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data*. Information Sciences **622**, 178–210, 2023. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2022.11.139.

[15] K. Kailing, H.-P. Kriegel, and P. Kröger: *Density-Connected Subspace Clustering for High-Dimensional Data*. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, 246–256. doi: 10.1137/1.9781611972740.23.

[16] F. Kaya and S. Şahin: *Comparison of Hierarchical and Non-Hierarchical Clustering Algorithms*. 2017. https://api.semanticscholar.org/CorpusID:212456885.

[17] J. Kleinberg: *An impossibility theorem for clustering*. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'02, 463–470. MIT Press, Cambridge, MA, USA, 2002.

[18] J. Li, C. Peng, W. Gu, and et al: *A simple oriented search and clustering method for extracting individual forest trees from ALS point clouds*. Ecological Informatics **86**, 102978, 2025. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2024.102978.

[19] G. Mei, C. Saltori, E. Ricci, and et al: *Unsupervised Point Cloud Representation Learning by Clustering and Neural Rendering*. International Journal of Computer Vision **132**, 3251–3269, 2024. doi: 10.1007/s11263-024-02027-5.

[20] S. Monti, P. Tamayo, J. Mesirov, and T. Golub: *Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data*. Machine Learning **52**, 91–118, 2003. doi: 10.1023/A:1023949509487.

[21] H.-S. Park and C.-H. Jun: *A simple and fast algorithm for K-medoids clustering*. Expert Systems with Applications **36**(2, Part 2), 3336–3341, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2008.01.039.

[22] P. J. Rousseeuw: *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics **20**, 53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7.

[23] N. Sadretdinova, L. Ivanova, J. Siegmund, and Y. Kyosev: *Methodology for Design of Well Fitting Load-Bearing Belts*. vol. 1, #36. 2024. doi: https://doi.org/10.15221/24.36.

[24] N. Sadretdinova, J. Siegmund, and Y. Kyosev: *Computational Design of an Adaptable Harness-Based Carrying System for Patients with Dementia Symptoms*. Fibres & Textiles in Eastern Europe **33**(1), 49–57, 2025. doi: 10.2478/ftee-2025-0005.

[25] R. Thorndike: *Who Belongs in the Family?* Psychometrika **18**, 267–276, 1953. doi: https://doi.org/10.1007/bf02289263.

[26] C. M. Weber, D. Ray, A. A. Valverde, J. A. Clark, and K. S. Sharma: *Gaussian mixture model clustering algorithms for the analysis of high-precision mass measurements*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **1027**, 166299, 2022. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2021.166299.

[27] S. Ying, G. Xu, C. Li, and Z. Mao: *Point Cluster Analysis Using a 3D Voronoi Diagram with Applications in Point Cloud Segmentation*. ISPRS International Journal of Geo-Information **4**(3), 1480–1499, 2015. ISSN 2220-9964. doi: 10.3390/ijgi4031480.

Received December 12, 2025; final form February 19, 2026.