

A New Partitioning Process for Geometrical Product Specifications and Verification

Na Cai^{a*}, Nabil Anwer^a, Paul J. Scott^b, Lihong Qiao^c, Xiangqian Jiang^b

^aLURPA, ENS Paris-Saclay, Univ. Paris-Sud, Université Paris-Saclay, 94235 Cachan, France

^bUniversity of Huddersfield, Huddersfield, U.K.

^cDepartment of Industrial and Manufacturing Systems Engineering, School of Mechanical Engineering & Automation, Beihang University, Beijing, 100191, China

Abstract

In ISO Geometrical Product Specifications and Verification Standards (GPS), Feature operations are used to obtain ideal and non-ideal features. The formalization of such operations enables to reduce ambiguity and uncertainty within the activities of design, manufacture and metrology of mechanical products, and their scientific investigation contributes to develop a sound mathematical framework and formalisms for the comprehension of engineering practices and the development of new standards. Partitioning is a fundamental operation defined in ISO GPS standard which aims at decomposing a part into independent features or surface portions for further processing and analysis. In this paper, a state-of-the-art survey of partitioning and segmentation methods and techniques reported in the literature is conducted and a comprehensive classification is proposed. Thereafter, a new partitioning process is developed for partitioning into regions and recognizing each region as one of the seven invariance classes of surfaces. It proceeds in three main steps: initial partitioning based on shape index and curvedness, refined partitioning by slippage analysis and invariance class recognition by statistical evaluation. An intuitive shape color wheel is defined to visualize the partitioned features according to their corresponding invariance classes. Experiments and results on both nominal models and measured point clouds are presented to demonstrate the effectiveness of the proposed method.

Keywords: Partitioning; Geometrical Product Specifications and Verification; Invariance classes; Curvature

1. Introduction

In ISO Geometrical Product Specifications and Verification Standards (GPS), Feature operations are used to obtain ideal and non-ideal features. The formalization of such operations enables to reduce ambiguity and uncertainty within the activities of design, manufacture and metrology of mechanical products, and their scientific investigation

* Corresponding author.

contributes to develop a sound mathematical framework and formalisms for the comprehension of engineering practices and the development of new standards. Among these operations Partitioning, Extraction, Filtration and Association have been particularly investigated. Partitioning is used to identify ideal or non-ideal features, Extraction enables to identify specific points from a non-ideal feature, Filtration is used to create the feature representing the considered characteristics from a non-ideal feature, and Association is used to fit ideal feature(s) to non-ideal feature(s) according to a criterion. Feature operations have been particularly researched in recent years by the research community from mathematical and computational point of view and some important results have been highlighted by new standardization efforts and the recent developments of geometry processing methods and tools in computational metrology.

Partitioning is one of the fundamental operations used to specify the geometry of a product. Partitioning operation is described as the process to divide an extracted skin model of a part into subsets, and each subset can be corresponded to a surface feature on the boundary of a nominal model of the ideal shape. Partitioning operations can also be processed on the boundary of a nominal model of ideal shape for ideal features definition concerning specification application [1]. Partitioning called also segmentation plays an important role in geometry processing and is addressed by different topics and fields such as medical imaging, computer graphics, civil engineering and mechanical engineering in the context of reverse engineering.

In ISO GPS, all ideal features are categorized into seven invariance classes according to their invariance degrees [2]. The invariance degree is described as the displacement(s) of the ideal features through which the ideal features are kept identical in the space. In order to obtain ideal or non-ideal features from skin model, nominal model or real part, specific operations are necessary. Four main feature operations, namely Extraction, Partitioning, Filtration and Association, are defined in ISO GPS standards.

Partitioning methods have been extensively investigated in amounts of literature. However, none of them can recognize the exact invariance class of each partitioned region, which makes the proceeding of further GPS operations difficult. On the other hand, each partitioning method has its own advantages and drawbacks, and combining the essence of these methods is the key to an improved performance. In this paper, we propose a hybrid partitioning process combining edge detection, attribute clustering, region growing and statistical evaluation for GPS. Three mathematical notions that are automorphism [3,4], slippable motion [5] and curvature are used to capture the invariance properties of a geometric set, which provide solid mathematical foundation for GPS-oriented partitioning.

The proposed process proceeds in three main steps: initial partitioning based on shape index and curvedness, refined partitioning by slippage analysis and invariance class recognition by statistical evaluation. The partitioned features are visualized using the defined shape color wheel according to the invariance classes that they belong to. Experimental results show that the proposed method can achieve not only appropriate partitioning but also accurate invariance class recognition for geometrical product

specifications and verification. It performs well for both nominal models and real parts. In addition, it also shows good robustness for partitioning low-density data and noisy data. Therefore, it is believed that this research can help the exploration and standardization of partitioning operations spearheaded by the ISO/TC 213/AG 12 (Mathematical Support Group).

The remainder of this paper is organized as follows: Section 2 summarizes the existing point cloud and mesh partitioning methods. Section 3 introduces the invariance classes defined in ISO GPS and the related partitioning methods. Section 4 introduces the curvature-based method. A hybrid partitioning process is proposed in Section 5. In Section 6, experiments are used to verify the method and the results are discussed. Section 7 draws the conclusions.

2. Point cloud and mesh partitioning methods

Various methods have been developed to partition 3D point cloud and mesh into homogeneous regions based on geometrical and topological criteria. Surveys have been conducted to review and categorize partitioning methods from different perspectives [6-11]. Anwer et al. [1] gave a general classification on current partition approaches, which includes edge detection, region growing, attribute clustering and hybrid approach, to provide scientific foundation for the development of ISO GPS partitioning standards. They pointed out that an appropriate classification was considered to be one of the prerequisites for the detailed standards development. In this paper, a more comprehensive and detailed classification is proposed. The existing partitioning methods are classified into seven main categories and subsequently some of them are further divided into sub-categories according to their core methodologies, as shown in Fig.1. Then all the categories are described and compared below.

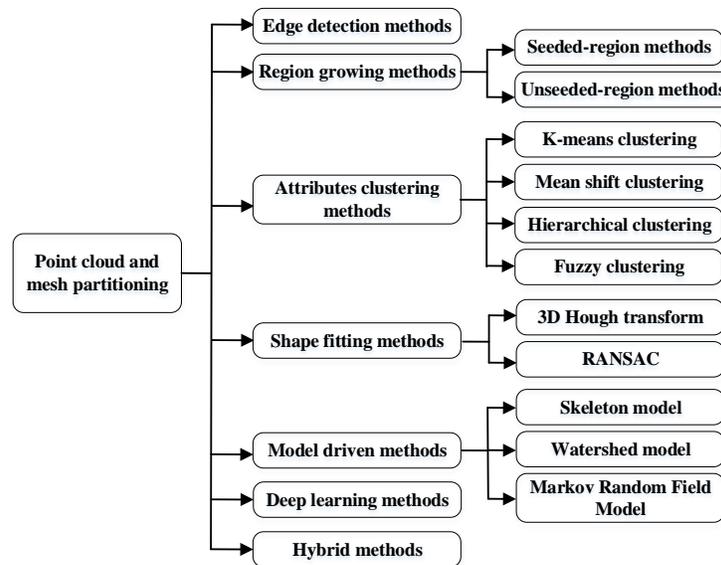


Fig. 1. Classification of point cloud and mesh partitioning methods

Edge detection methods realize partitioning by detecting the edges to outline the borders of different regions and then group points inside the boundaries [12]. Bhanu et al. [13] presented an edge detection approach by calculating the gradient, fitting 3D lines to points and detecting changes in unit normal vector direction. Wani and Arabnia [14] defined and extracted three types of edges from the equidepth contours obtained from sliced 3D images. It is noticed that edge detection methods often detected disconnected edges in 3D space, which made the identification of closed segments difficult [15].

Partitioning by region growing is achieved by combining points in neighborhood that have similar attributes until a set of termination or growing criteria is satisfied, and then different regions distinguished by dissimilarity are obtained. Depending on the existence of seed points, it is classified into seeded-region (or bottom-up) methods and unseeded-region (or top-down) methods [7]. Sometimes, triangle is selected as seed and region is aggregated by triangle [16]. Some improved region growing methods are also developed such as the two-phase octree-based region growing method [17].

In attributes clustering methods, the points are clustered into partitioned regions by evaluating their attributes. The results depend on the selected attributes and their derivation techniques as well as the clustering algorithm. Four clustering algorithms are commonly used for partitioning: K-means clustering [18,19], mean shift clustering [20,21], hierarchical clustering [22,23] and fuzzy clustering [24,25]. The attributes are selected according to the domain of interest, and the commonly used ones include curvature, convexity, normal, automorphism, slippage, etc.

Shape fitting methods try to fit primitive shapes such as planes, spheres, cylinders, cones and tori from the point cloud. Points belonging to the same primitive shape are labeled as one partitioned region [26]. Vosselman et al. extended the classical Hough transformation in image segmentation to three dimensional point partitioning [27, 28]. Schnabel et al. [29, 30] used RANSAC method to extract shapes by randomly drawing minimal sets from point data and constructing corresponding shape primitives.

Model driven methods assume that objects in a point cloud or a mesh are presented in a certain pattern. Skeleton model extracts the skeleton of the object and use it to guide the partitioning [31,32]. Watershed model is a mathematical morphological approach similar to the way that water fills a geographic surface. The points where flood regions meet as the water floods its basins construct the watershed lines that divide the surface into different regions [33,34]. Markov random field (MRF) model is a probabilistic graphical model used to find the optimal labeling of the nodes of a graph [35]. Based on MRF model, each vertex of the mesh is assigned the most correct label and then clustered into different regions [36].

Recent advances in machine learning, especially deep learning technologies, have revolutionized 2D image segmentation and recognition with outstanding performance. Now the revolution has also been targeted on 3D point cloud and mesh. Shu et al. [37] developed an unsupervised algorithm for shape partitioning using deep learning. Guo et al. [38] proposed a 3D mesh partitioning method based on deep Convolutional Neural

Networks (CNN). Although 3D point cloud and mesh has a more explicit and accurate representation of object, its complex structure makes its deployment in deep architectures not simple [39]. Besides, the emerged deep learning methods for partitioning are mainly applied on natural objects at present, the application on CAD parts needs to be developed.

Each partitioning method mentioned above has its own advantages and drawbacks. Therefore, these different methods are sometimes fused to improve overall performance. Roggero [40] combined the region growing techniques with principal component analysis (PCA) for object partitioning of data set from airborne laser scanners. Vieira and Shimada [41] developed a hybrid method integrating region growing and shape fitting methods to partition laser range scanned data. Yi, et al. [42] combined tensor voting method, region growing, attribute clustering method and RANSAC method to recognize primitive shapes. Zhao and Anwer [43, 44] combined edge detection method and attribute clustering method for mesh segmentation.

Table 1 Summary and comparison of partitioning methods

Partitioning methods	References	Advantages	Disadvantages	Shape recognition
Edge detection methods	[12] [13] [14] [15]	High efficiency	Insufficient robustness; Inaccurate results in case of noise and uneven density of point clouds	✗
Region growing methods	[16] [17]	Robustness to noise	Sensitivity to the initial seed selection, the attributes used and the termination criteria, etc.	✗
Attributes clustering methods	[18] [19] [20] [21] [22] [23] [24] [25]	High efficiency	Sensitivity to the noise in the data and the definition of neighborhood; the results depend on the selected attributes and their derivation techniques	✗
Shape fitting methods	[26] [27] [28] [29] [30]	High efficiency; robustness to outliers	Difficulty in determining the initial minimal point sets; Incapability for complex shapes	Geometric primitives
Model driven methods	[31] [32] [33] [34] [35] [36]	Being intuitive to use the existing models	High computational complexity, having limitation on 3D point clouds, e.g. over segmentation	✗
Deep learning methods	[37] [38] [39]	High accuracy and robustness	High computational cost	✗
Hybrid methods	[40] [41] [42] [43] [44]	Improved performance	✗	Depend on combined methods

All the seven categorizations of partitioning methods mentioned above are summarized in Table 1 and their advantages and disadvantages are compared. It can be seen that none of the partitioning methods can recognize the exact shape of each segmented region except

for shape fitting methods which can only identify some geometric primitives such as plane, cylinder, etc. However, ideal features defined in ISO GPS are classified according to the invariance classes. Therefore, it is critical to recognize the invariance classes of the partitioned regions for the implementation of the ISO GPS standards.

In this paper, a hybrid partitioning process is proposed to partition point cloud and mesh into regions, and to recognize the exact invariance class of each region. Finally, a shape color wheel is defined for quick and accurate visual identification of the seven invariance classes.

3. Invariance class related partitioning methods

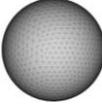
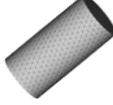
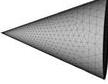
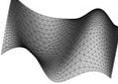
According to ISO GPS, all ideal features can be categorized into one of the seven invariance classes, and the invariance attributes can be captured by the notions of automorphism and slippable motion. Gelfand and Guibas [5] used slippage signature analysis for region partitioning, but they didn't identify the invariance class that each region belonged to. Chiabert, et al. [45-48] proposed a statistical model to evaluate the invariance of a geometric set. It can be used to recognize the specific invariance class of each region.

3.1 Invariance classes in ISO GPS

As shown in Table 2, ISO GPS standards define seven invariance classes that cover all kinds of ideal features [2]. An automorphism of a geometric set S is defined as a group of rigid motions that leaves S invariant in the Euclidean space, denoted as $Aut(S) = \{m \in M : m(S) = S\}$, where m is rigid motion and M is the set of all rigid motions. The connected component of $Aut(S)$ that contains the identity is denoted as $Aut_0(S)$ [3]. When we use automorphism to define the invariance of a geometric set S , S is usually considered as a continuous surface. Since any continuous surface can be approximated by a triangular mesh represented by three topological elements that are vertices, edges and triangle facets, automorphism can also be used to describe the invariance of a triangular mesh. It is worth noticing that for precise approximation of continuous surface, the topological elements of a triangular mesh should be dense enough. Table 2 lists some examples of triangular mesh and their corresponding invariance attributes.

Pottmann et al. [49] also proposed the concept of uniform equiform motion to describe the invariance attributes. They defined that an equiform kinematic surface is a surface traced out by a curve undergoing a uniform equiform motion [50] and classified the equiform kinematic surfaces into four categories.

Table 2 ISO GPS invariance classes [1-3]

Number	Invariance class	Discrete differential geometry example S	$Aut_0(S)$	Reference element or tuple	$\dim(Aut_0(S))$
S1	Spherical		$R(3)$	Point	3
S2	Cylindrical		$T(1) \times R(1)$	Straight line	2
S3	Planar		$T(2) \times R(1)$	Plane	3
S4	Helical		$T(1) \times R(1)$ with pitch μ	Helix	2
S5	Revolute		$R(1)$	(Point, Straight line)	1
S6	Prismatic		$T(1)$	(Straight line, Plane)	1
S7	Complex		i	(Point, Straight line, Plane)	0

3.2 Statistical evaluation of invariance classes

Corresponding to the seven invariance classes, seven semi-parametric models $M_i (i = 1, \dots, 7)$ define the structure of Probability Density Functions (PDFs) having the symmetries. These models are built upon the set of parameters R_i related to the reference elements or tuple of invariance classes, and the projection function $F_i(S; R_i)$ which replaces the original set S with its projection on the quotient set \mathfrak{R} / E_i (where E_i is the set of equivalent points by automorphism motions). The seven semi-parametric models are listed in Table 3, and more details can be referred to [45-48].

Table 3 The seven semi-parametric models and their transformed PDFs [45-48]

Semi-parametric models	Reference parameters	Transformation of co-ordinate reference system	Transformed PDF
The spherical model M_1	Sphere center $R_1 = \{o_x, o_y, o_z\}$	$(x, y, z) \rightarrow (\rho, \varphi, \theta)$	$p(\rho, \varphi, \theta M_1, R_1, F_1(S; R_1)) = \frac{\cos(\theta)}{4\pi} p^\rho(\rho F_1(S; R_1))$
The cylindrical model M_2	Symmetry axis $R_2 = \{r_x, r_y, t_x, t_y\}$	$(x, y, z) \rightarrow (\rho, \varphi, z')$	$p(\rho, \varphi, z' M_2, R_2, F_2(S; R_2)) = \frac{1}{2\pi} p^\rho(\rho F_{2\rho}(S; R_2)) p^{z'}(z' F_{2z'}(S; R_2))$
The planar model M_3	Normal vector $R_3 = \{n_x, n_y, n_z\}$	$(x, y, z) \rightarrow (x', y', z')$	$p(x', y', z' M_3, R_3, F_3(S; R_3)) = p^{x'}(x' F_{3x'}(S; R_3)) p^{y'}(y' F_{3y'}(S; R_3)) p^{z'}(z' F_{3z'}(S; R_3))$
The helical model M_4	Symmetry axis and helix pitch $R_4 = \{r_x, r_y, t_x, t_y, \mu\}$	$(x, y, z) \rightarrow (\rho, h, z')$	$p(\rho, h, z' M_4, R_4, F_4(S; R_4)) = p^{\rho h}(\rho, h F_{4\rho h}(S; M_4)) p^{z'}(z' F_{4z'}(S; R_4))$
The revolute model M_5	Symmetry axis $R_5 = \{r_x, r_y, t_x, t_y\}$	$(x, y, z) \rightarrow (\rho, \varphi, z')$	$p(\rho, \varphi, z' M_5, R_5, F_5(S; R_5)) = \frac{1}{2\pi} p^{\rho z'}(\rho, z' F_5(S; R_5))$
The prismatic model M_6	Normal vector $R_6 = \{n_x, n_y, n_z\}$	$(x, y, z) \rightarrow (x', y', z')$	$p(x', y', z' M_6, R_6, F_6(S; R_6)) = p^{x'y'}(x', y' F_{6x'y'}(S; R_6)) p^{z'}(z' F_{6z'}(S; R_6))$
The complex model M_7	$R_7 = \emptyset$	$(x, y, z) \rightarrow (x, y, z)$	$p(x, y, z M_7, R_7, F_7(S; R_7)) = p(x, y, z S)$

3.3 Slippage analysis method

The invariance of a geometric set S can also be captured by the notion of slippable motion, under which the velocity vector of each vertex $\mathbf{p}_i \in S$ is tangent to S at \mathbf{p}_i [5]. If a triangular mesh can be approximated by an invariance surface, we deem it to be slippable. In order to calculate its slippable motions, the motion along the normal direction at each vertex is minimized as:

$$\min_{[\mathbf{r} \ \mathbf{t}]} \sum_{i=1}^n ((\mathbf{r} \times \mathbf{p}_i + \mathbf{t}) \cdot \mathbf{n}_i)^2 \quad (1)$$

where, \mathbf{r} is a rotation vector, \mathbf{t} is a translational vector, \mathbf{p}_i is the coordinates of the vertex and \mathbf{n}_i is the normal at the vertex.

Eq.(1) is a least-squares problem, which can be solved by a linear system $M_c \mathbf{x} = 0$. M_c is a covariance matrix of second partial derivatives of the objective function with respect to the motion parameters.

$$M_c = \sum_{i=1}^n \begin{bmatrix} m_{ix}m_{ix} & m_{ix}m_{iy} & m_{ix}m_{iz} & m_{ix}n_{ix} & m_{ix}n_{iy} & m_{ix}n_{iz} \\ m_{iy}m_{ix} & m_{iy}m_{iy} & m_{iy}m_{iz} & m_{iy}n_{ix} & m_{iy}n_{iy} & m_{iy}n_{iz} \\ m_{iz}m_{ix} & m_{iz}m_{iy} & m_{iz}m_{iz} & m_{iz}n_{ix} & m_{iz}n_{iy} & m_{iz}n_{iz} \\ n_{ix}m_{ix} & n_{ix}m_{iy} & n_{ix}m_{iz} & n_{ix}n_{ix} & n_{ix}n_{iy} & n_{ix}n_{iz} \\ n_{iy}m_{ix} & n_{iy}m_{iy} & n_{iy}m_{iz} & n_{iy}n_{ix} & n_{iy}n_{iy} & n_{iy}n_{iz} \\ n_{iz}m_{ix} & n_{iz}m_{iy} & n_{iz}m_{iz} & n_{iz}n_{ix} & n_{iz}n_{iy} & n_{iz}n_{iz} \end{bmatrix} \quad (2)$$

where $m_{ik} = (\mathbf{p}_i \times \mathbf{n}_i)_k$, $k = x, y, z$

Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_6$ be the eigenvalues of M_c and $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_6$ be the corresponding eigenvectors. The slippable motions of a triangular mesh are eigenvectors whose corresponding eigenvalues are zero. However, M_c could be full rank due to noise. Therefore, the slippage motions are those eigenvectors of M_c whose eigenvalues are small enough as:

$$\lambda_6 / \lambda_i > th_{se} \quad i = 1, \dots, k \quad (3)$$

where k is the number of small eigenvalues, and th_{se} is a user-determined threshold.

The corresponding eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ are called the slippage signature of \mathbf{p}_i , and they are represented in a $6 \times k$ matrix as $SS = [\mathbf{v}_1, \dots, \mathbf{v}_k]$.

Gelfand and Guibas [5] used slippage analysis for region partitioning and they pointed out that the algorithm was most sensitive to the chosen initial patch size. Besides, they didn't identify the specific invariance class that each region belonged to. Yi et al. [42] also proved the impact of initial patch size on partitioning. If the initial patch size is too small, the patch may be incorrectly recognized as high slippage shapes since small patch usually looks like plane even if it is a cylinder of large radius. On the other hand, if the initial patch size is too large, the patch may be incorrectly recognized as low slippage shapes. Improper initial patch size also causes wide edges or disconnected edges. However, it is difficult to determine the most appropriate initial patch size analytically. The adopted solution in the literatures is iterative try out and adjustment of initial patch size by evaluating the partitioning quality, which evidently decreases the efficiency and robustness of the algorithm.

4. Curvature-based partitioning method

Curvature is an important tool for analyzing the geometry of surfaces because it specifies the second-order properties of a surface and is independent of the used parametrization. A surface can be approximated by a triangular mesh, and accurate

definition and derivation of discrete curvatures is essential for high quality approximation. Meek and Walton [51] summarized three classes of discrete curvature estimation methods: discretization of curvature operators defined for smooth surface [52], approximation of a local quadric surface on a given vertex to the mesh and calculation of the discrete curvature by applying the derivatives [53]; and tensor-based techniques to estimate discrete curvature [54]. In this paper, we use the first method for discrete curvature estimation.

4.1 Definitions of shape index and curvedness

In this paper, two notions of shape index and curvedness [55] are applied to describe the local shape of surface quantitatively and intuitively.

Shape index, ranged $[-1, 1]$, specifies the local shape type in the neighborhood of a vertex \mathbf{p}_i . It is independent from size and the assignment of principal directions. As a single shape indicator, shape index captures the notion of the local shape well and its value scale can be intuitively categorized into distinct shapes. The shape index of a vertex \mathbf{p}_i is calculated as:

$$s(\mathbf{p}_i) = \frac{-2}{\pi} \arctan\left(\frac{\kappa_1(\mathbf{p}_i) + \kappa_2(\mathbf{p}_i)}{\kappa_1(\mathbf{p}_i) - \kappa_2(\mathbf{p}_i)}\right) \quad (\kappa_1 \geq \kappa_2) \quad (4)$$

where $\kappa_1(\mathbf{p}_i)$ and $\kappa_2(\mathbf{p}_i)$ are the maximum and minimum principal curvature of a vertex \mathbf{p}_i , respectively. The principal curvatures are obtained from the shape operator which is also called Weingarten endomorphism [56].

Curvedness, always positive, specifies the size, which is the amount of the surface curvatures in the neighborhood of a vertex \mathbf{p}_i . It is calculated as:

$$c(\mathbf{p}_i) = \frac{\sqrt{(\kappa_1^2(\mathbf{p}_i) + \kappa_2^2(\mathbf{p}_i))}}{2} \quad (5)$$

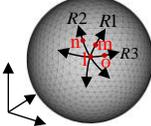
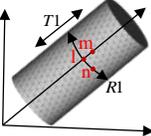
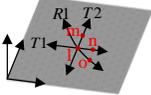
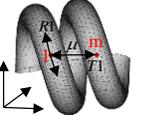
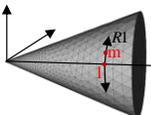
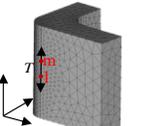
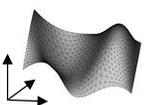
The pair of shape index and curvedness contains the information equivalent to the pair of maximum and minimum principal curvatures or the pair of Gaussian and mean curvatures. It is more intuitive by decoupling the size and shape: the shape index describes the shape independently of size while the curvedness describes the size.

4.2 Invariance attribute of curvature

Table 4 lists the curvatures of the invariance classes. It can be seen that for each invariance class the curvature is in a certain range. The spherical, cylindrical and planar classes have invariant shape index and curvedness values. As a special case of the revolute class, cone also has invariant shape index and curvedness values which are $s = \pm 0.5$, $c = |\kappa_1|/2$ or $c = |\kappa_2|/2$ ($\kappa_2, \kappa_1 = const$). Therefore, the spherical class, cylindrical class,

planar class and cone which are commonly used primitive shapes in engineering can be robustly recognized by curvature.

Table 4 The curvatures of the invariance classes [1-3]

Invariance class	Discrete differential geometry example	$Aut_0(S)$	Maximum and minimum principal curvature κ_1, κ_2	Shape index and curvedness	Invariance of curvature
Spherical		$R(3)$	$\kappa_1 = \kappa_2 = const$	$s = \pm 1,$ $c = \kappa_1 /\sqrt{2} = \kappa_2 /\sqrt{2}$	$\kappa_l = \kappa_m = \kappa_n = \kappa_o$ $s_l = s_m = s_n = s_o$ $c_l = c_m = c_n = c_o$
Cylindrical		$T(1) \times R(1)$	$\kappa_1 = const, \kappa_2 = 0$ or $\kappa_1 = 0,$ $\kappa_2 = const$	$s = \pm 0.5, c = \kappa_1 /2$ or $c = \kappa_2 /2$	$\kappa_l = \kappa_m = \kappa_n$ $s_l = s_m = s_n$ $c_l = c_m = c_n$
Planar		$T(2) \times R(1)$	$\kappa_1 = \kappa_2 = 0$	$c = 0$	$\kappa_l = \kappa_m = \kappa_n = \kappa_o$ $s_l = s_m = s_n = s_o$ $c_l = c_m = c_n = c_o$
Helical		$T(1) \times R(1)$ with pitch μ	$\kappa_1 \neq 0, \kappa_2 \neq 0$	$s \in [-1, 1], c > 0$	$\kappa_l = \kappa_m$ $s_l = s_m$ $c_l = c_m$
Revolute		$R(1)$	$\kappa_1 \geq \kappa_2$	$s \in [-1, 1], c > 0$	$\kappa_l = \kappa_m$ $s_l = s_m$ $c_l = c_m$
Prismatic		$T(1)$	$\kappa_1 > 0, \kappa_2 = 0$ or $\kappa_1 = 0, \kappa_2 < 0$	$s = \pm 0.5, c = \kappa_1 /2$ or $c = \kappa_2 /2$	$\kappa_l = \kappa_m$ $s_l = s_m$ $c_l = c_m$
Complex		i	$\kappa_1 \geq \kappa_2$	$s \in [-1, 1], c > 0$	0

In addition, the curvature of a vertex has invariance attribute along the automorphism or slippable direction. Take the spherical class in Table 4 for example. Rotation $R1$ is an automorphism motion of the sphere example S , that is $R1 \in M$, so according to the definition of automorphism we can obtain $R1(S) = S$. Since the l and m are two vertices along rotation $R1$ direction, we can obtain $R1(l) = l = m$. In this case, the curvatures of vertices l and m are the same too. That is, $\kappa_l = \kappa_m, s_l = s_m, c_l = c_m$. Therefore, it's not difficult to deduce that the curvatures of the vertices along the automorphism or slippable direction are the same.

We can use the invariance attribute of curvature for invariance class-oriented partitioning. In practice, the curvatures of the vertices along the automorphism or slippable direction cannot be exactly the same because of the curvature estimation error due to noise and the definition of neighborhood. Therefore, shape index, ranged $[-1, 1]$, is divided into multiple value intervals and in each value interval a distinct shape can be determined. And the planar class can be determined by the value of curvedness.

4.3 Partitioning based on shape index and curvedness

The shape index and curvedness of each vertex are evaluated at first, and then partitioning is achieved by identifying and clustering all the vertices into sharp edges and ten shape types according to their shape index and curvedness values [57, 58].

Sharp edges are just transitions between two surfaces and they are transition features. Therefore, vertices belonging to sharp edges should be detected at first. In practice, the curvedness of a vertex at a sharp edge is usually much larger than other vertices. Therefore, it can be identified when the curvedness is larger than a given threshold c_e :

$$E(\mathbf{p}) = \{\mathbf{p}_i | c_i - c_{\min} > c_e\} \quad (6)$$

where $E(\mathbf{p})$ is the collection of vertices that locate on sharp edges, and \mathbf{p}_i is one arbitrary vertex in $E(\mathbf{p})$. c_i is the curvedness of \mathbf{p}_i ; c_{\min} is the minimum curvedness of all the vertices and c_e is a given threshold calculated by:

$$c_e = \delta \cdot \left(\frac{c_{\max} + c_{\min}}{2} \right) \quad (7)$$

where c_{\max} is the maximum curvedness of all the vertices and δ is a user-determined coefficient ($\delta = 1$ by default). If some sharp edges are not detected, δ should be set lower so that the threshold c_e is lower to identify the undetected edges. On the other hand, if some vertices that do not belong to sharp edges are detected as sharp edges, δ should be set higher so that the threshold c_e is higher to avoid mistakenly detecting vertices.

After sharp edges detection, the other vertices are identified as one of the predefined ten local shape types according to their shape index and curvedness value. The value of curvedness is always positive except for an ideal plane vertex whose curvedness is zero. In practice, the curvedness of a plane vertex cannot be exactly zero due to noise and evaluation error, but is a small value. Therefore, the plane shape is identified when the curvedness is less than a given threshold c_p (c_p is a small value, for example: $c_p = 1 \times 10^{-5}$). The other nine shape types are determined according to the shape index range. The sharp edges and ten shape types are visualized in pre-assigned colors as shown in Fig.2.

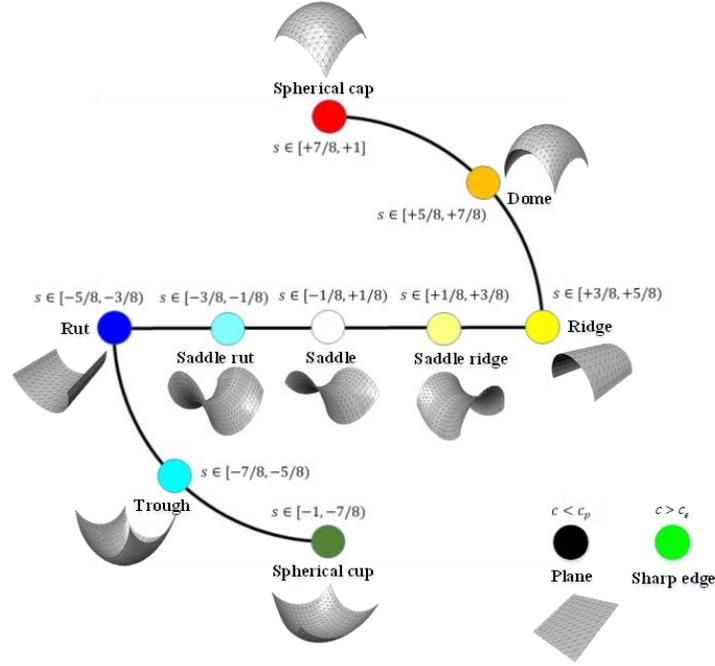


Fig. 2. Illustration of shape index and curviness range of ten shape types and sharp edge and their visualization colors (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The categorization of local shapes into ten shape types is reasonable because the discretization of shape index and curviness value scale is fine enough to distinguish the invariance classes. Besides, the ten shape types are visually meaningful and can be easily discriminated.

The curvature-based partitioning method is straightforward. However, since the curvature of individual vertex is usually estimated using vertices in local neighborhood, the curvature-based partitioning method is sensitive to the noise in the data and the definition of neighborhood.

5. Invariance class-oriented hybrid partitioning process

In Section 3.3 we have explained that by randomly decomposing the point cloud P into initial patches $P_1, \dots, P_i, \dots, P_n$, the partitioning quality is influenced by the size of the initial patch greatly. However, it is difficult to determine the appropriate initial patch size analytically. To solve the problem, curvature-based method is applied to form the proper initial patches as the input of the slippage analysis method for partitioning.

The shape index and curviness of each vertex are calculated and then all the vertices are clustered into regions according to their shape types which is determined by the shape index and curviness value range. Due to the invariant attribute of curvature, vertices in the same region have the same invariance, hence the regions are slippable unless it belongs to the complex class. These regions become the input initial patches of the slippage analysis

method and their slippage signatures can be correctly evaluated. This guarantees good partitioning quality in the following steps.

It is worth noting that both curvature-base method and slippage analysis method can only partition point cloud into regions with the same shape types or slippage signatures. However, none of the two methods can identify the exact invariance class that each region belongs to. Therefore, in this paper a statistical evaluation method is used to recognize the exact invariance class of each region after partitioning.

5.1 Framework

The framework of the proposed hybrid partitioning process is illustrated in Fig.3. The input is point cloud and mesh acquired by tessellation of nominal CAD model or from measurement. The output is partitioned features and each feature is recognized as one of the seven invariance classes defined in ISO GPS. It is proceeded in three main steps: 1) initial region partitioning based on shape index and curvedness, 2) refined region partitioning by slippage analysis and 3) invariance class recognition by statistical evaluation.

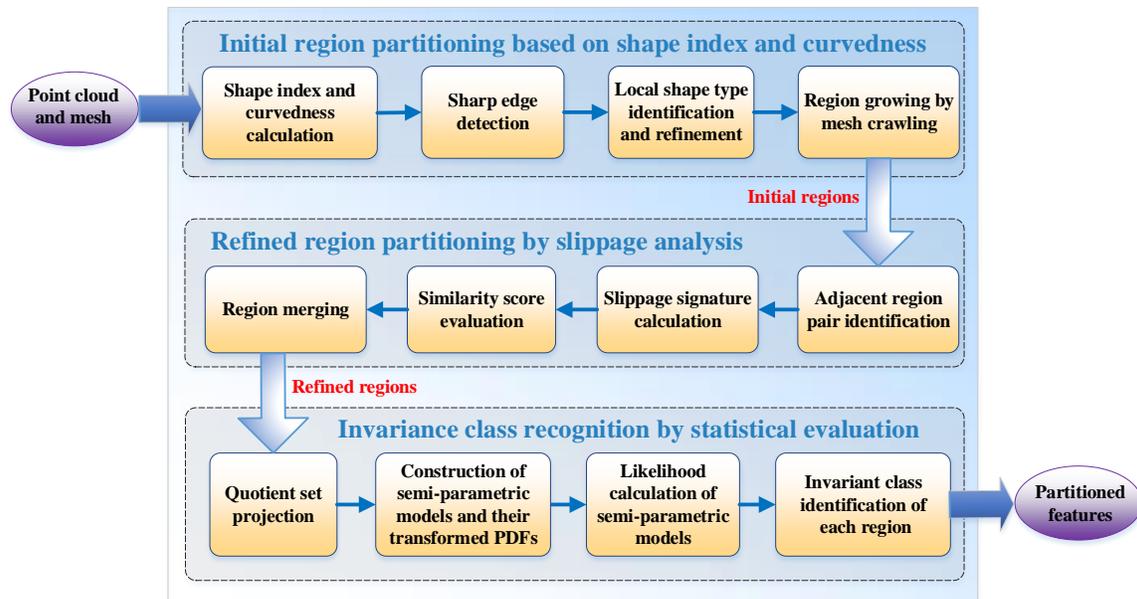


Fig. 3. Framework of the hybrid partitioning process

In the first step, the shape index and curvedness of each vertex are calculated and the vertices are clustered into sharp edge and ten local shape type clusters by evaluating the range of shape index and curvedness. An iterative voting algorithm is implemented to refine the local shape types of vertices. Finally, the shape type clusters are partitioned into isolated regions using region growing method by mesh crawling.

The regions obtained in the first step becomes the input of the refined region partitioning. Adjacent region pairs are identified by checking topological neighborhood. Slippage signatures of regions are calculated and the similarity score of each adjacent region pair is

evaluated. The adjacent regions having high similarity score are merged into one region. The merging process iterates until all adjacent region pairs have similarity scores lower than a given threshold. Thereafter, the refined regions are obtained.

Every refined region is considered as a geometric set composed of points which can be projected on the quotient set with equivalent points by automorphism motion. Seven semi-parametric models corresponding to the seven invariance classes are constructed. Then the original three dimensional Probability Density Functions (PDFs) are transformed according to the invariance hypothesis. By calculating and ranking the likelihoods of the seven semi-parametric models, it can be determined that the model with largest value of likelihood is the best fitting model. Hence, the region is identified as the corresponding invariance class of this model.

After the above three steps, point cloud and mesh can be precisely partitioned into separated features and each feature is recognized as one of the seven invariance classes in ISO GPS. In the following sub-sections, each step will be explained in detail.

5.2 Initial region partitioning based on shape index and curvedness

As mentioned in Section 4.3, two curvature-based attributes that are shape index and curvedness are evaluated for each vertex first, and then the vertices are clustered into sharp edges and ten shape types. For convenient and straightforward representation of the local shape types, a series of integers $\{-4, -3, \dots, 4, 5\}$ are assigned for the local shape types according to their shape index and curvedness value range, formulized as follows:

$$T(\mathbf{p}_i) = \begin{cases} \left\lfloor \frac{\lfloor 8s \rfloor}{2} \right\rfloor & c \geq c_p, s \in [-1, 1] \\ 5 & c < c_p \end{cases} \quad (8)$$

where, $T(\mathbf{p}_i)$ is the local shape type label of vertex \mathbf{p}_i . $\lfloor x \rfloor$ represents floor function (the largest integer less than or equal to x), while $\lceil x \rceil$ represents ceiling function (the smallest integer greater than or equal to x).

Table 5 lists the ten shape types identified by shape index and curvedness and their corresponding integer representations calculated by equation (8).

Table 5. Shape types and the corresponding integer representations.

Shape Type	Shape index and curvedness	Integer representation
Spherical cup	$s \in [-1, -7/8)$	-4
Trough	$s \in [-7/8, -5/8)$	-3
Rut	$s \in [-5/8, -3/8)$	-2
Saddle rut	$s \in [-3/8, -1/8)$	-1
Saddle	$s \in [-1/8, +1/8)$	0

Saddle ridge	$s \in [1/8, 3/8)$	1
Ridge	$s \in [3/8, 5/8)$	2
Dome	$s \in [5/8, 7/8)$	3
Spherical cap	$s \in [7/8, 1]$	4
Plane	$c < c_p$	5

After local shape type identification, an iterative voting method is applied to refine the local shape types of all the vertices. Finally, the shape type clusters are partitioned into isolated regions using region growing method.

5.2.1 Local shape type refinement

The local shape types of vertices can be ideally identified for tessellated mesh model generated from nominal CAD model. However, for the mesh model built from measured point data, the local shape types may be incorrectly identified due to various errors, such as the manufacturing errors of surface, the measurement errors of point data and the mathematical errors from surface approximation and discrete curvature evaluation, etc.

To solve the problem, an iterative voting method is applied to refine the local shape types of vertices according to their neighborhood information. In each iteration, given a vertex \mathbf{p}_i and its neighbor vertices (eg. one-ring stencil) $N(\mathbf{p}_i)$, we can calculate the shape type difference between \mathbf{p}_i and any neighbor vertex $\mathbf{p}_j \in N(\mathbf{p}_i)$ by Eq. (9).

$$d(\mathbf{p}_j, \mathbf{p}_i) = \begin{cases} \frac{1}{8} \cdot |T(\mathbf{p}_j) - T(\mathbf{p}_i)| & T(\mathbf{p}_i) \neq 5, T(\mathbf{p}_j) \neq 5 \\ 0 & T(\mathbf{p}_i) = 5, T(\mathbf{p}_j) = 5 \\ \frac{1}{9} & T(\mathbf{p}_i) = 5, T(\mathbf{p}_j) \neq 5 \end{cases} \quad (9)$$

where $T(\mathbf{p}_i)$ and $T(\mathbf{p}_j)$ are the local shape types of \mathbf{p}_i and \mathbf{p}_j , respectively.

It can be explained by the (κ_1, κ_2) plane shown in Fig.4. The direction of a half-ray reflects the local shape type while the distance from the origin reflects the local size of a vertex. Moreover, the origin represents planar vertex. When the shape type difference is calculated, size is not considered. So the shape types of all vertices can be projected on the same circle, and the difference of shape types can be evaluated by the angle between the two half-rays as shown in Fig.4. Four situations are considered for calculating the shape type difference between two vertices:

- Both the shape types of \mathbf{p}_i and \mathbf{p}_j are non-planar: the nine non-planar shape types are evenly distributed on the circle as shown in Fig.4. Assume the largest shape

type difference is 1 (between $T = 4$ and $T = -4$), then the shape type difference between any two vertices can be calculated as $\frac{1}{8} \cdot |T(\mathbf{p}_j) - T(\mathbf{p}_i)|$.

- Both the shape types of \mathbf{p}_i and \mathbf{p}_j are planar: the shape types are the same, so the shape type difference is 0.
- The shape type of \mathbf{p}_i is planar and \mathbf{p}_j is non-planar: the shape type difference between planar and any of the nine non-planar shape types are the same and set as 1/9.
- The shape type of \mathbf{p}_i is non-planar and \mathbf{p}_j is planar: since planar shape type is identified and adjusted by curvedness, so we don't try to adjust non-planar shape type to planar shape type. In this case, we will ignore the influence of \mathbf{p}_j on shape type refinement.

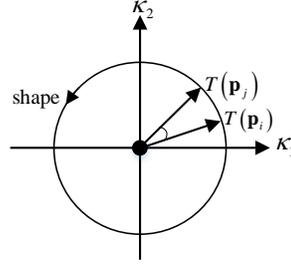


Fig. 4. Shape type difference illustration on (κ_1, κ_2) plane

The possibility that the local shape type of \mathbf{p}_i should be relabeled to that of its neighboring vertex \mathbf{p}_j can be evaluated by Eq. (10) and it should be refined to the one with maximum possibility, as shown in Eq.(11). The iteration terminates when the local shape types of all vertices do not change any more or a convergence condition is met.

$$P_i[T(\mathbf{p}_j)] = \frac{[1 - d(\mathbf{p}_j, \mathbf{p}_i)] \cdot n_j}{\sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} [1 - d(\mathbf{p}_j, \mathbf{p}_i)] \cdot n_j} \quad (10)$$

where n_j is the number of vertices with local shape type $T(\mathbf{p}_j)$ in the neighborhood $N(\mathbf{p}_i)$.

$$T(\mathbf{p}_i) = \max_{\mathbf{p}_j \in N(\mathbf{p}_i)} \{P_i[T(\mathbf{p}_j)]\} \quad (11)$$

5.2.2 Region growing by mesh crawling

After sharp edge detection and local shape type identification, the vertices are clustered into eleven different shape type clusters. In each cluster, the local shape types of the vertices are the same but the vertices may not be connected. Therefore, the connected vertices are group together as one region using region growing method by mesh crawling. In this way, a shape type cluster is partitioned into several regions.

A surface part example is used to explain the method, and its local shape types are visualized in Fig.5. It can be seen that the vertices are clustered into 3 shape type clusters: sharp edge cluster (lime), plane cluster (black) and ridge cluster (yellow). It's evident from Fig.5 that the plane cluster contains two isolated regions. In order to explain and display the region growing process clearly, here we use coarse mesh so that the grid size is large enough for visualization, as shown in Fig.6. First of all, a vertex in the plane cluster is randomly chosen as the seed. Then the vertices that connect with the seed vertex are found based on the topological relationship. The search for connected vertices continues until all the vertices that have connective relationship are found. In this way, the isolated region is generated. The whole process is like mesh crawling starting from seed vertex toward radial outward direction until no more connected vertex can be found. The region generation process is iterated until all shape clusters are partitioned into isolated regions.

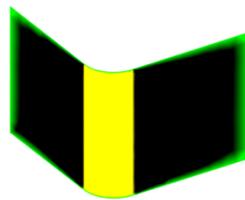


Fig. 5. Shape type visualization of a surface part example (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

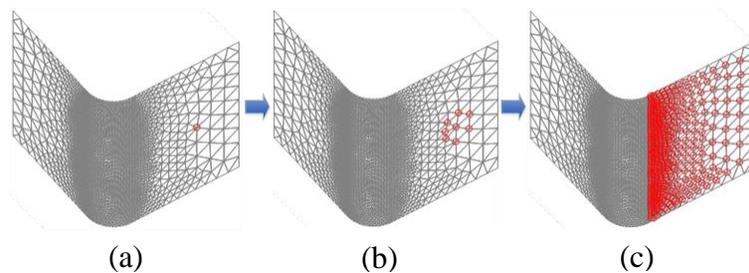


Fig. 6. Illustration of region growing: (a) seed vertex choosing; (b) connected vertices searching; (c) isolated region generation

Finally, the planar shape type cluster (shown as black in Fig.5) is partitioned into two isolated regions (shown as red and yellow in Fig.7) and the whole surface is segmented into four regions in total, as displayed in different colors in Fig.7. The sharp edge is shown in dark blue. The transition between region and sharp edge is displayed by gradient color. The sharp edge looks wide because the grid size is large along the sharp edge.

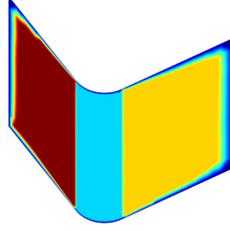


Fig. 7. Region visualization of the surface part example (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The initial regions obtained by curvature-based partitioning belong to the defined sharp edge and the ten local shape types. However, the ten local shape types can hardly convey invariance properties. Therefore, the initially obtained regions will be classified into the invariance classes based on slippage analysis in the next section. In addition, even with shape type refinement, some vertices can still be ill-identified due to the inaccurate estimation of curvature when there is noise. This will lead to some small improperly-partitioned regions. In the next step, slippage analysis will be used to merge these improperly-partitioned regions into their neighboring regions with high similarity score.

5.3 Refined region partitioning by slippage analysis

The problem of partitioning by slippage analysis lies in the difficulty of determining the most appropriate initial patch size analytically. To solve this problem, point cloud or mesh is firstly partitioned based on shape index and curvedness. Then the obtained regions become the initial patches as the input of slippage signature-based partitioning. The vertices in each initial patch belong to the same local shape type, which makes the calculation of slippage signature more accurate. Meanwhile, sharp edges are detected by curvedness and won't be influenced by the initial patch size. After slippage signature-based partitioning, the regions are refined and each of them belongs to one of the invariance classes.

For each initial patch, we can form a covariance matrix M_c , and the slippage signature is calculated as the eigenvectors of small eigenvalues of M_c . The partitioning process proceeds by merging adjacent patches with high similarity score into slippable components. First, we identify the adjacency relationship between patches and form adjacent patch pairs. For mesh input, if two patches are connected by edges they are considered as adjacent patch pair. For point cloud input, if two patches share points they are adjacent patch pair.

Consider each adjacent patch pair (P_i, P_j) , their slippage signatures are SS_i and SS_j . Then their similarity score is computed as:

$$Sim(P_i, P_j) = G(\sigma_{k+1}) \cdot C(P_i, k) \cdot C(P_j, k) \quad (12)$$

where k is the minimum number of slippable motions considered, σ_{k+1} is the $(k+1)$ st singular value of the combined matrix of slippage signatures $[SS_i \ SS_j]$ and G is the Gaussian function centered around 0 with its width determining the difference degree between the slippage signatures of adjacent patch pair (P_i, P_j) . The function C acts as a confidence multiplier for the similarity score:

$$C(P_i, k) = \begin{cases} 1 & \text{if } \frac{\lambda_6}{\lambda_k} \geq th_{se} \text{ and } \frac{\lambda_6}{\lambda_{k+1}} < th_{se} \\ 1 & \text{if } \frac{\lambda_6}{\lambda_{k+1}} \geq th_{se} \text{ and } |P_i| < N \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where N is the desired size of individual segmented components.

The adjacent patches with high similarity score are merged into one region. The similarity evaluation and region merging process iterates for 3 times (from $k=3$ to $k=1$) until all the adjacent regions have low similarity score [5]. The obtained regions belong to the invariance classes, whereas the regions with no slippable motion ($k=0$) belong to complex class.

5.4 Invariance class recognition by statistical evaluation

After refined region partitioning based on slippage signature, the invariance class that each region belongs to should be determined. Each region is a geometric set S composed of a collection of vertices $\{\mathbf{p}_1, \dots, \mathbf{p}_i, \dots, \mathbf{p}_n\}$, where $\mathbf{p}_i = (x_i, y_i, z_i)$. It can be represented by the indicator function i_s :

$$i_s : \mathfrak{R}^3 \rightarrow \{0,1\} \quad \forall \mathbf{p} \in \mathfrak{R}^3, i_s(\mathbf{p}) = 1 \Leftrightarrow \mathbf{p} \in S \quad (14)$$

Then the indicator function i_s inherits the invariance attribute of S in automorphism motions as described in Section 3. Each invariance class in Table 2 admits a probabilistic model M_i having some peculiar structure derived from the symmetries in S . The invariance class recognition process is explained as follows.

A region $S = \{\mathbf{p}_1, \dots, \mathbf{p}_i, \dots, \mathbf{p}_n\}$ where $\mathbf{p}_i = (x_i, y_i, z_i)$ can be considered as a sample extracted from a PDF whose parametric form is unknown. Therefore, seven transformed PDFs based on the seven invariance probabilistic models $M_i (i=1, \dots, 7)$ are reconstructed for the region S , as shown in Table 3. Then the fitting degree between the region S and the reconstructed PDFs is used to discover the symmetries in region. The likelihood of model M_i is calculated by Leave-One-Out method [48] as

$$\hat{L}(M_i) = \prod_{j=1}^n \tilde{p}(x_j, y_j, z_j | M_i, R_{ij}, F_i(S_j; R_{ij})) \quad (15)$$

$$S_j = S \setminus \{(x_j, y_j, z_j)\} \quad \forall j \in \{1, \dots, n\} \quad (16)$$

$$R_{ij} = \arg \max_{R_i} \tilde{p}(S_j | M_i, R_i, F_i(S_j; R_i)) \quad (17)$$

where R_{ij} is the maximum likelihood estimate of minimum reference parameters R_i for S_j .

In practice, we use $\log \hat{L}(M_i)$ to evaluate the fitting degree. The model with the largest value of $\log \hat{L}(M_i)$ is considered as the best fitting model, and the region S is considered to belong to the corresponding invariance class.

After invariance class recognition, quick and accurate identification of features is meaningful and important for geometrical product specifications and verification. Therefore, in this paper a shape color wheel is developed to visualize features according to their invariance classes, as shown in Fig.8.

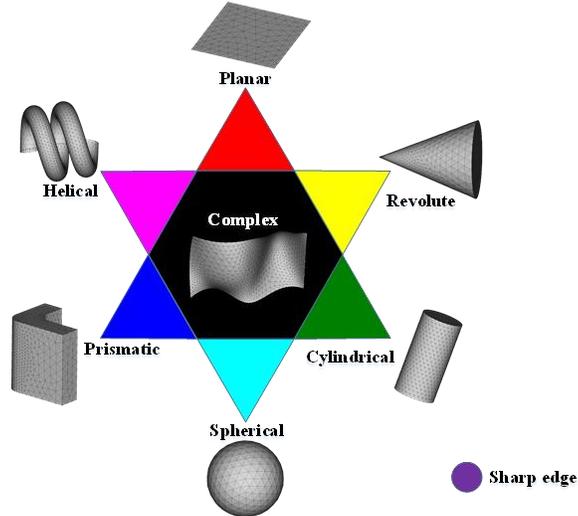


Fig. 8. Shape color wheel for the invariance classes (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The main principle is that the colors should be easily recognizable and nameable. Six colors (red, yellow, green, cyan, blue and magenta) evenly distributed on the color circle are used to represent the six invariance classes with $\dim(\text{Aut}_0(S)) > 0$ (planar, revolute, cylindrical, spherical, prismatic and helical). Features with the same dimension of automorphism are represented by complementary colors. Complex invariance class with $\dim(\text{Aut}(S)) = 0$ stands out by black color. Sharp edges are visualized in purple.

6. Experiments and results

The hybrid partitioning process proposed in this paper is implemented in MATLAB R2017a environment. Various nominal models built in CATIA V5 and measured point clouds have been tested. The results of the proposed partitioning process has also been compared with the curvature-based method and the slippage analysis method on low-density data, high-density data as well as noisy data.

6.1 Nominal models

A case study part composed of semi-sphere and cube (we call it sphere-cube part) is used to explain and verify the proposed methodology. In the initial partitioning step, sharp edges and shape types of case study model are identified and visualized as shown in Fig.9(a), and then 59 isolated regions are partitioned as shown in Fig.9(b). Fig.9(c) shows the refined regions obtained by slippage signature-based partitioning. The 59 initial regions are merged into 28 regions and each refined region belongs to one type of the invariance class. The color is randomly chosen to distinguish each region in Fig.9 (b) and (c).

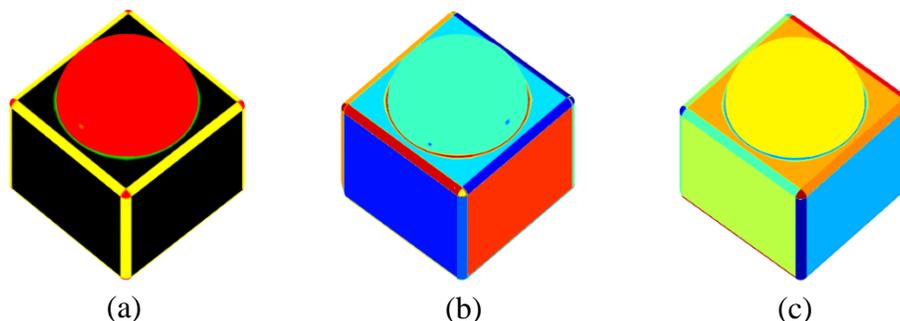


Fig. 9. Partitioning of the sphere-cube part (a) Shape types; (b) initial regions (c) refined regions (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Thereafter, the invariance class of each region is recognized by the statistical model. Three regions obtained by the refined region partitioning method and their corresponding rankings of $\log \hat{L}(M_i)$ against invariance classes are shown in Fig.10-12. It can be seen that the three regions are correctly recognized as planar, cylindrical and spherical classes, respectively. Since helical shape is not commonly used in engineering, it's not considered in the case study.

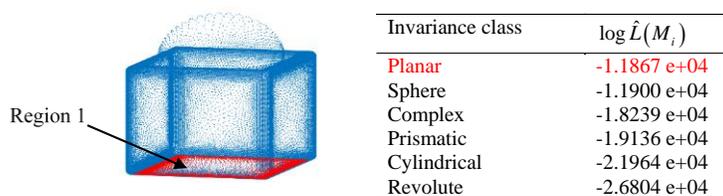


Fig. 10. Region 1 and its rankings of $\log \hat{L}(M_i)$ against invariance classes

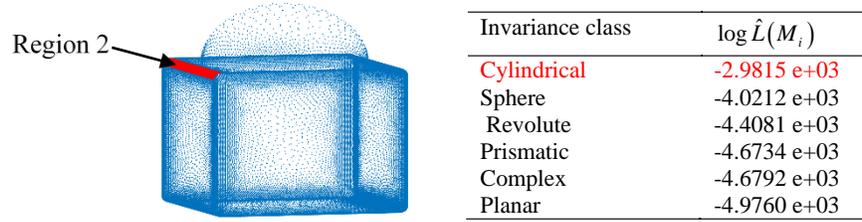


Fig. 11. Region 2 and its rankings of $\log \hat{L}(M_i)$ against invariance classes

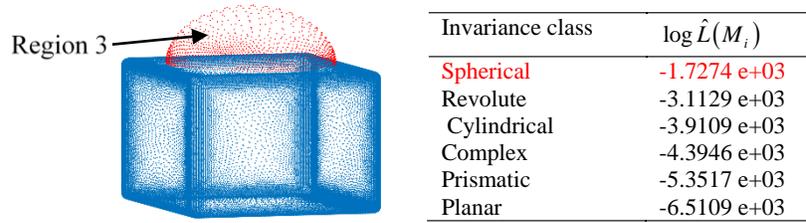


Fig. 12. Region 3 and its rankings of $\log \hat{L}(M_i)$ against invariance classes

Finally, the features of the case study model are recognized and visualized according to the invariance classes that they belong to, as shown in Fig.13. Features belonging to three invariance classes that are planar, cylindrical and spherical are correctly recognized and visualized in red, green and cyan, respectively. The sharp edge between sphere and plane is shown in purple color.

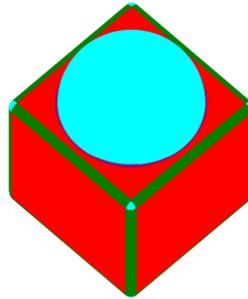


Fig. 13. Invariance class recognition of the sphere-cube part (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We have also tested our hybrid partitioning process on other nominal models. Fig.14 shows the results of partitioning and recognition for a simple part. First of all, the nominal CAD model is tessellated into mesh model and vertices in mesh model are clustered into sharp edges and ten shape types by computing shape index and curvedness shown in Fig.14(a). Then it is partitioned into isolated regions shown in Fig.14(b). Because the part has a simple shape and discrete curvatures are accurately estimated, all vertices are correctly identified and partitioned into 20 regions including 19 surface regions and 1 sharp edge region. In this case, region refinement by slippage signature-based partitioning is even

unnecessary. Finally, based on statistical evaluation, the exact invariance class of each region is recognized and visualized by the defined shape color wheel, as shown in Fig.14(c).

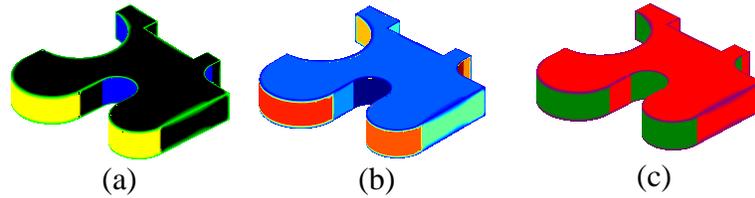


Fig. 14. Partitioning results for a simple part. (a) Curvature-based shape identification; (b) Region segmentation (20 regions); (c) Invariance class recognition (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig.15 shows the results of partitioning and recognition for a revolution part example. Sharp edges and shape types are identified as shown in Fig.15(a). It is obvious that some vertices are incorrectly identified, especially in torus area. This leads to the unsatisfying partitioning results of total 48 regions with some small regions as shown in Fig.15(b). Therefore, a slippage signature-based partitioning method is applied to refine the regions. The small regions are correctly merged into the neighborhood regions and the region number is refined into 16, including 6 sharp edge regions and 10 surface regions, as shown in Fig.15(c). The invariance class of each region is recognized and visualized in Fig.15(d).

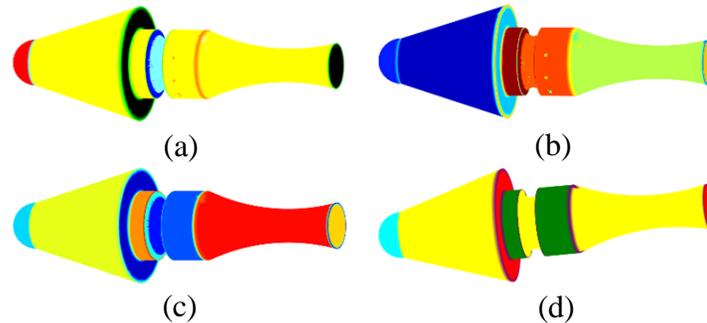


Fig. 15. Partitioning results for a CAD part. (a) Curvature-based shape identification. (b) Initial region segmentation (48 regions). (c) Slippage signature-based region partitioning (16 regions). (d) Invariance class recognition (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig.16 shows the results of partitioning and recognition for a fandisk model. From left to right, the three columns give the back, right and front view of the fandisk. In Fig.16(a), the mesh model is identified into sharp edges and ten defined shape types based on shape index and curvedness and then it's partitioned into 252 regions as shown in Fig.16(b). Next, the 252 regions are refined into 25 regions by using slippage signature-based partitioning, which includes 24 surface regions and 1 sharp edge region, as shown in Fig.16(c). The invariance classes are recognized as shown in Fig.16(d).

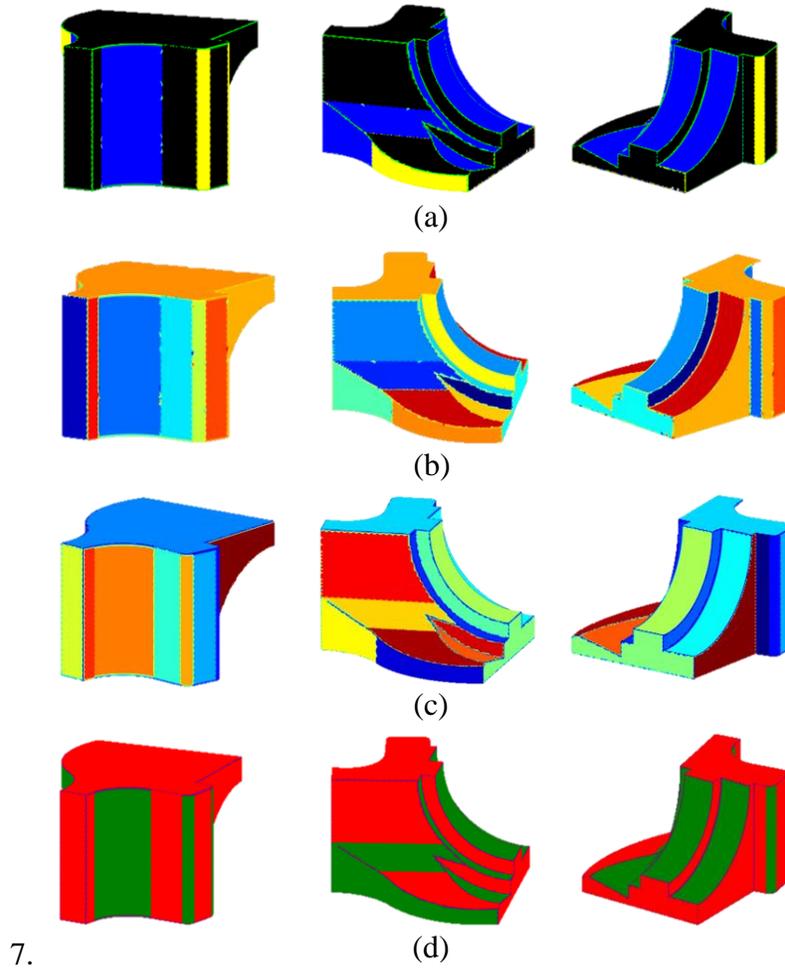


Fig. 16. Partitioning results for a fandisk model. (a) Curvature-based shape identification. (b) Initial region segmentation (252 regions). (c) Slippage signature-based region partitioning (25 regions). (d) Invariance class recognition (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.1 Point cloud

The proposed partitioning process is also tested on measured point cloud of a blade, as shown in Fig.17(a). At first, the point cloud is meshed to build the topology of the shape so that discrete curvatures can be estimated. Then sharp edges and ten shape types are identified for all the vertices. The initial identification result is not so satisfactory due to various errors, as shown in Fig.17(b). Therefore, the local shape types are refined by the proposed iterative voting method, as shown in Fig.17(c). However, there are still some vertices with local shape type incorrectly identified, leading to many small regions in the initial region partitioning, as shown in Fig.17(d). After slippage signature-based region partitioning, small regions are merged and the number of regions is greatly decreased so that the segmentation quality is significantly improved, as shown in Fig.17(e). The invariance classes are recognized as shown in Fig.17(f).

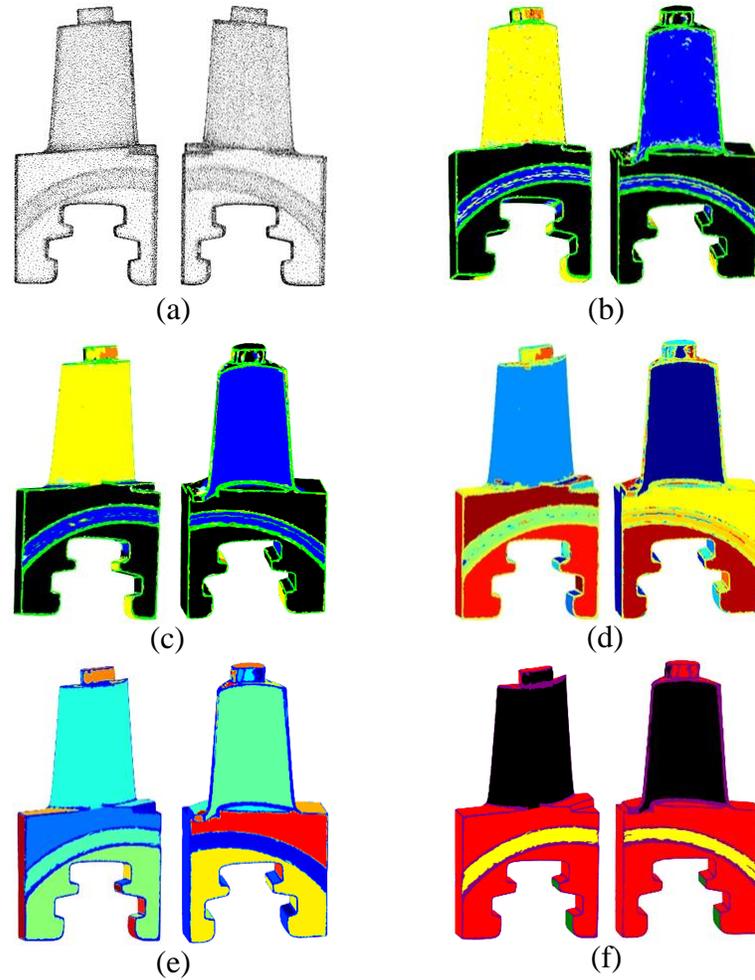


Fig. 17. Partitioning results for point cloud of a blade. (a) Measured point cloud of a blade; (b) Local shape type identification; (c) Local shape type refinement; (d) Initial region partition (1102 regions); (e) Slippage signature-based region partitioning (48 regions); (f) Invariance class recognition (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.2 Comparisons

The curvature-based partitioning, the slippage analysis-based partitioning and the proposed hybrid partitioning are tested and compared on various data including a low-density nominal data with 3520 vertices, a high-density nominal data with 12472 vertices and a low-density noisy data with 3520 vertices. The partitioning results are shown in Fig.18.

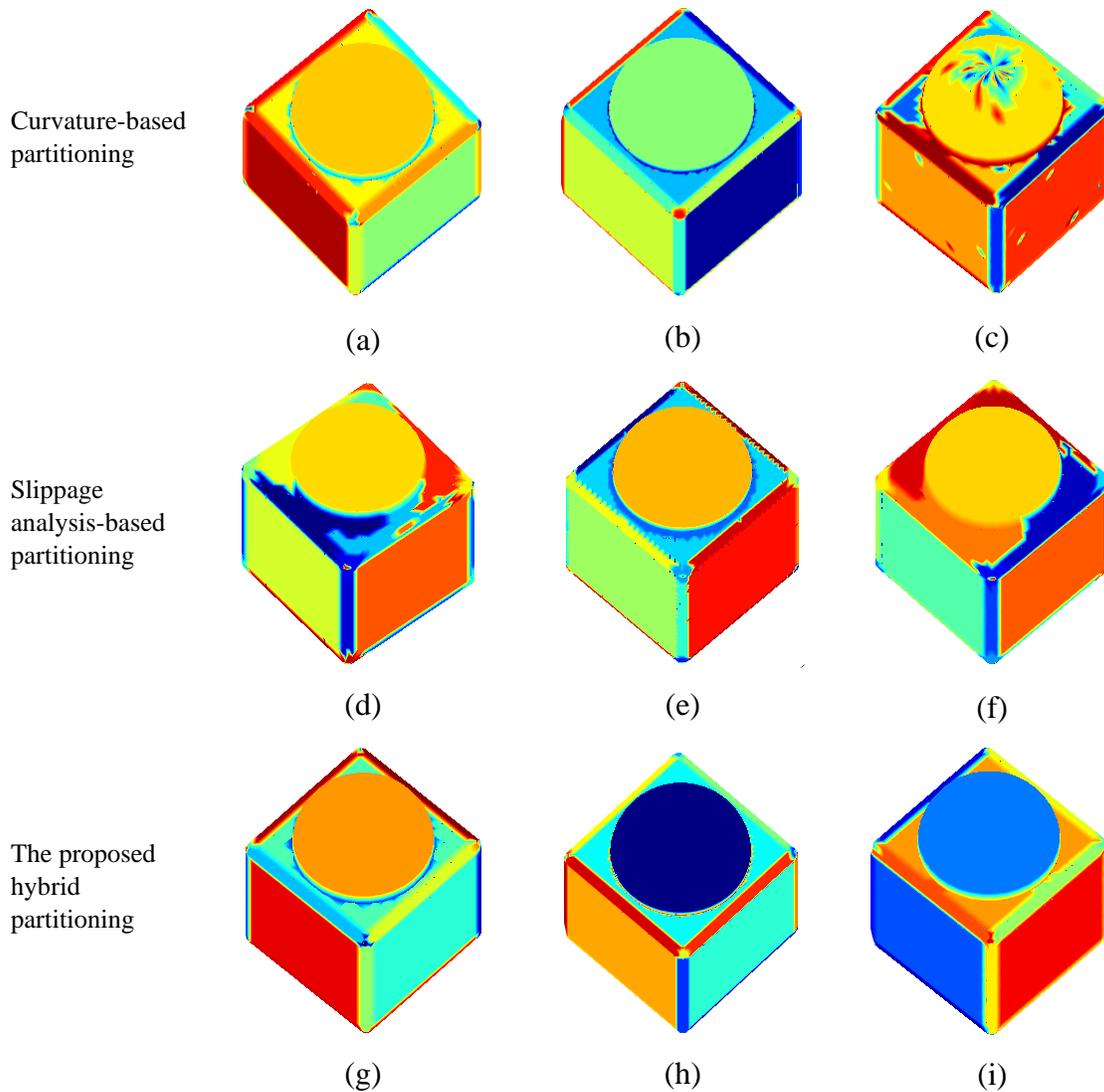


Fig.18 Partitioning result comparison between the curvature-based partitioning, the slippage analysis-based partitioning and the proposed hybrid partitioning (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

It can be seen that all the methods achieve better partitioning on the high-density nominal data than on the other two sets of data. It also shows that the results obtained by the hybrid partitioning process are the best. In the curvature-based partitioning, the curvature of a vertex is estimated approximately by its local neighborhood, so it is sensitive to the description of the neighborhood. Therefore, the curvatures of some vertices may not be accurately evaluated and then their shape types may not be correctly identified, which leads to incorrect small region generation. In the slippage analysis-based partitioning, the iterative adjustment of initial patch size increases the calculation amount of partitioning process evidently. Moreover, the edges between two regions identified by the slippage analysis method are not as smooth as those identified by the other two methods.

The performance of the curvature-based partitioning is affected greatly by the noise. Due to the influence of the noise, the evaluation of curvature is inaccurate and it then leads to many small regions in partitioning, as shown in Fig.18(c). The slippage analysis-based partitioning has poor performance on low density data. It is because that the low density can cause the inaccurate estimation of slippage signature and then lead to wrong patch growing, especially for transition area. It can be also seen that the proposed hybrid partitioning process has better performances than the curvature-based partitioning and the slippage analysis-base partitioning in all the situations. It is robust to both low density and noise situation.

The results above show that the proposed hybrid partitioning process can achieve not only appropriate partitioning but also accurate invariance class recognition for geometrical product specifications and verification. The defined shape color wheel facilitates the visualization of invariance classes for quick and accurate identification. In addition, the proposed partitioning process performs well for both nominal models and measured point clouds. And it shows better robustness compared with the curvature-based partitioning and the slippage analysis-based partitioning in the situations of low-density data and noisy data.

8. Conclusion

In this paper, a state-of-the-art survey of existing partitioning methods is performed and a comprehensive and detailed classification of partitioning methods is proposed. Subsequently a hybrid partitioning process for geometrical product specifications and verification is developed. The method proceeds in three main steps: initial partitioning by evaluating shape index and curvedness, refined partitioning based on slippage analysis and invariance class recognition by statistical evaluation. Through these steps, the shape is partitioned into regions and each region is recognized as one of the seven invariance classes defined in ISO GPS. Finally a shape color wheel is defined to visualize regions according to their invariance classes.

The proposed hybrid partitioning process has been tested on various models and point clouds. It has been proved that the proposed method can partition both tessellated nominal models and measured point clouds appropriately and it has good robustness in the situations of low density and noise of the vertices. In addition, it enables the accurate invariant class recognition for geometrical product specifications and verification.

Therefore, it paves the way and draws new type of classification and usage for partitioning to enrich the concepts and provide scientific basis for ISO 18183 series spearheaded by the ISO/TC 213/AG 12 (Mathematical Support Group). Thus, default partitioning and geometric specifications can be addressed considering invariance classes of surfaces which is a fundamental classification principle in ISO GPS.

A future work will investigate new statistical approaches for vertex clustering, robust estimation of normal and curvature parameters, and the evaluation of partitioning results. Furthermore, the newly emerging and promising geometric deep learning methods will be investigated to improve the proposed method [59].

Acknowledgement

The authors thank Vijay Srinivasan from National Institute of Standards and Technology, USA for very useful suggestions. We also wish to acknowledge support from the ISO/TC 213/AG 12 (Mathematical Support Group).

References

- [1] Anwer N, Scott PJ, Srinivasan V. Toward a classification of partitioning operations for standardization of geometrical product specifications and verification. *Procedia CIRP* 2018; 75: 325-330.
- [2] ISO 17450-1:2011. Geometrical product specifications (GPS) – General concepts – Part 1: Model for geometrical specifications and verification. International Organization for Standardization. Geneva, 2011.
- [3] Srinivasan V. A geometrical product specification language based on a classification of symmetry groups. *Comput Aided Des* 1999; 31(11): 659-668.
- [4] Srinivasan V. *Theory of dimensioning: An introduction to parameterizing geometric modes*. New York: Marcel Dekker; 2004.
- [5] Gelfand N, Guibas LJ. Shape segmentation using local slippage analysis. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*; 2004. p. 214-223.
- [6] Shamir A. A survey on mesh segmentation techniques. *Comput Graph Forum* 2008; 27(6): 1539-1556.
- [7] Nguyen A, Le B. 3D point cloud segmentation: A survey. In: *6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*; 2013. p. 225-230.
- [8] Grilli E, Menna F, Remondino F. A review of point clouds segmentation and classification algorithms. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2017; 42(2): 339-344.
- [9] Theologou P, Pratikakis I, Theoharis T. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. *Comput Vis Image Underst* 2015; 135: 49-82.
- [10] Rodrigues RSV, Morgado JFM, Gomes AJP. Part-Based Mesh Segmentation: A Survey. *Comput Graph Forum* 2018; 1-40
- [11] Agathos A, Pratikakis I, Perantonis S, et al. 3D mesh segmentation methodologies for CAD applications. *Comput Aid Des Applic* 2007; 4(6): 827-841.
- [12] Rabbani T, Van Den Heuvel F, Vosselmann G. Segmentation of point clouds using smoothness constraint. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2006; 36(5): 248-253.
- [13] Bhanu B, Lee S, Ho C C, et al. Range data processing: Representation of surfaces by edges. In: *Proceedings of the IEEE International Pattern Recognition Conference*; 1986. p. 236-238.
- [14] Wani MA, Arabnia HR. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. *J Supercomput* 2003; 25(1): 43-62.
- [15] Castillo E, Liang J, Zhao H. Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. In: *Innovations for Shape Analysis*, Berlin: Springer; 2013. p. 283-299.

- [16] Lavoué G, Dupont F, Baskurt A. Constant curvature region decomposition of 3d meshes by a mixed approach vertex-triangle. *J WSCG* 2004; 12 (2): 245-252.
- [17] Vo A V, Truong-Hong L, Laefer D F, Bertolotto M. Octree-based region growing for point cloud segmentation. *ISPRS J Remote photogramm Remote Sens* 2015; 104: 88-100.
- [18] Lavoué G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis. *Comput Aided Des* 2005; 37(10): 975-987.
- [19] Tatiraju S, Mehta A. Image Segmentation using k-means clustering, EM and Normalized Cuts. University Of California Irvine. 2008.
- [20] Yamauchi H, Lee S, Lee Y, Ohtake Y, Belyaev A, Seidel HP. Feature sensitive mesh segmentation with mean shift. In: *International Conference on Shape Modeling and Applications*; 2005. p. 236-243.
- [21] Zhang X, Li G, Xiong Y, He F. 3D mesh segmentation using mean-shifted curvature. In: *International Conference on Geometric Modeling and Processing*; 2008. p. 465-474.
- [22] Garland M, Willmott A, Heckbert P S. Hierarchical face clustering on polygonal surfaces. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*; 2001. p.49-58.
- [23] Lu X, Yao J, Tu J, Li K, Li L, Liu Y. 2016. Pairwise Linkage for Point Cloud Segmentation. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci* 2016; 3(3): 201-208.
- [24] Filin S. Surface clustering from airborne laser scanning data. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2002; 34:117–124.
- [25] Biosca JM, Lerma JL. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS J Photogramm Remote Sens* 2008; 63(1): 84-98.
- [26] Schnabel R, Wessel R, Wahl R, Klein R. Shape Recognition in 3D Point-Clouds. In: *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in co-operation with EUROGRAPHICS*; 2008. p. 65-72.
- [27] Vosselman G, Dijkman S. 3D building model reconstruction from point clouds and ground plans. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2001; 34(3): 37-44.
- [28] Vosselman G, Gorte BGH, Sithole G, Rabbani T. Recognising structure in laser scanner point clouds. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2004; 46(8): 33-38.
- [29] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection. *Comput Graph Forum* 2007; 26(2): 214-226.
- [30] Schnabel, R., Wahl, R., Klein, R., 2006. Shape detection in point clouds. Universitaet Bonn; Technical report No. CG-2006/2. ISSN 1610-8892.
- [31] Verroust A, Lazarus F. Extracting skeletal curves from 3D scattered data. *Visual Computer* 2000; 16:15-25.
- [32] Au OKC, Tai CL, Chu HK, D Cohen-Or, TY Lee. Skeleton extraction by mesh contraction. *ACM Trans Graph* 2008; 27(3): 44:1-44:10.
- [33] Karvelis PS, Tzallas AT, Fotiadis D I, Georgiou I. A Multichannel Watershed-Based Segmentation Method for Multispectral Chromosome Classification. *IEEE Trans Med Imaging* 2008; 27(5): 697-708.

- [34] Koschan A F. Perception-based 3D triangle mesh segmentation using fast marching watersheds. In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2003.
- [35] Lavoué G, Wolf C. Markov random fields for improving 3d mesh analysis and segmentation. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval; 2008. p. 25-32.
- [36] Anguelov D, Taskar B, Chatalbashev V, Koller D, Gupta D, Heitz G, et al. Discriminative learning of Markov random fields for segmentation of 3D scan data. In: IEEE conference on computer vision and pattern recognition (CVPR); 2005: p. 169-176.
- [37] Shu Z, Qi C, Xin S, Hu C, Wang L, Zhang Y, et al. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Comput Aided Geom Des* 2016; 43(C), 39-52.
- [38] Guo K, Zou D, Chen X. 3D mesh labeling via deep convolutional neural networks. *ACM Trans Graphic* 2015; 35(1): 3:1–3:12.
- [39] Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I. Deep learning advances in computer vision with 3D data: a survey. *J ACM Comput Surv* 2017; 50(2): 20:1-20:38
- [40] Roggero M. Object segmentation with region growing and principle component analysis. *Int Arch Photogramm Remote Sens Spatial Inf Sci* 2002; 34(3/A): 289-294
- [41] Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing. *Comput aided geom des* 2005; 22(8): 771-792.
- [42] Yi B, Liu Z, Tan J, Cheng F, Duan G, Liu L. Shape recognition of CAD models via iterative slippage analysis. *Comput Aided Des* 2014; 55: 13-25.
- [43] Anwer N, Yang Y, Zhao H, Coma O, Paul J. Reverse engineering for NC machining simulation. In: Proceedings of IDMME - Virtual Concept 2010; 2010: p.1-8.
- [44] ZHAO Haibin. Multisensor integration and discrete geometry processing for coordinate metrology. PhD thesis of Ecole Normale Superieue de Cachan, 2010
- [45] Chiabert P, Costa M. Probabilistic description of mechanical surfaces. In: Proceedings of 3rd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering; 2002. p. 479-484.
- [46] Chiabert P, Costa M. Probabilistic evaluation of invariant surfaces through the Parzen's method. *Geometric Product Specification and Verification: Integration of Functionality* 2003; 245-254.
- [47] Chiabert P, Costa M. Statistical modeling of nominal and measured mechanical surfaces. *J Comput Inf Sci Eng* 2003; 3:87-94.
- [48] Chiabert P, De Maddis M. Statistical modelling of geometrical invariant sampled sets. *Models for Computer Aided Tolerancing in Design and Manufacturing* 2007; 169-178.
- [49] Odehnal B., Pottmann H, Wallner J. Equiform kinematics and the geometry of line elements. *Contributions to Algebra and Geometry* 2006; 47(2): 567-582.
- [50] Hofer M, Odehnal B, Pottmann H, Wallner J. 3D shape recognition and reconstruction based on line element geometry. In: Tenth IEEE International Conference on Computer Vision (ICCV'05); 2005. p.1532-1538.
- [51] Meek D, Walton D. On surface normal and Gaussian curvature estimations given data sampled from a smooth surface. *J Comput Aided Geome Des* 2000; 17: 521-43.

- [52] Meyer M, Desbrun M, Schroder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics, Berlin: Springer; 2002, p. 35-57.
- [53] T. Srinark. Lecture note on differential geometry; 2008.
- [54] Cohen-Steiner D, Morvan JM. Restricted Delaunay Triangulation and Normal Cycle. In: Proceedings of 19th Annual ACM Symposium on Computational Geometry; 2003: p. 312-21.
- [55] Koenderink J J, Van Doorn A J. Surface shape and curvature scales. Image vision comput 1992; 10(8): 557-564.
- [56] O'neill B. Elementary Differential Geometry. New York: Academic Press; 1997
- [57] Bendjebba S, Cai N, Anwer N, Lavernhe S, Mehdi-Souzani C. Freeform Machining Features: New Concepts and Classification. Procedia CIRP 2018; 67: 482-487.
- [58] Cai N, Bendjebba S, Lavernhe S, Mehdi-Souzani C, Anwer N. Freeform Machining Feature Recognition with Manufacturability Analysis. Procedia CIRP 2018; 72: 1475-1480.
- [59] Bronstein M M, Bruna J, LeCun Y, Szlam, A, Vandergheynst, P. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine 2017; 34(4): 18-42.