



Reconstruction and Deformation of 3D Complex Models Based on Semantic Parameters

Kunjin He¹  and Xia Sun² 

^{1,2}HoHai University, kjinhe@163.com

Corresponding author: Kujin He, kjinhe@163.com

Abstract. To quickly redesign the 3D complex models including free-form surfaces, a method is proposed based on semantic parameters which facilitates the deformation of such models. To extend the model types, by calculating Euclidean distance between feature descriptors that represent the component attributes, the smallest Euclidean distance thus-obtained is used as a standard to choose the model that is the most similar to the target component from a set of components. Parametric deformation regards deformation handles as basic elements, and the hierarchical deformation can be realized by setting the semantic parameters: the 3D models can be redesigned with the deformation of components by configuring the deformation handle. First, the freeform surfaces of each component are replaced by the quadric surfaces with Hausdorff distance and the cluster analysis method. Then, the constraints of the internal and external components are calculated with the relationship between deformation handles and the relation to vertices of each component. Finally, the 3D model can be deformed based on editing and modifying the semantic parameters which are defined on the deformation handles. Experimental results which are verified through 3D vehicle models show that the above methods support users to achieve the deformation of 3D models with a few parameters and keep better efficiency of custom design.

Keywords: feature descriptor; cluster analysis; quadric surface; component constraint; semantic parameter

DOI: <https://doi.org/10.14733/cadaps.2019.664-678>

1 INTRODUCTION

Currently, in order to meet the needs of the public and pursue the stable sales of the products, the designers are designing products based on the aesthetic tendency and performance requirements in terms of most people. With the growing trend of product homogenization, people prefer incorporating their ideas into product design to accepting the existing products provided by company. However, product design requires that designers are not only familiar with the underlying surface modeling technology, but also proficient in geometric operation skills.

Therefore, in view of the existing 3D models, the user-specified model is refactored by utilizing the feature attributes of the 3D component model to enrich the product categories, and a replacement method of the free-form surfaces and the constraints among different components are explored. The goal of the above methods is to support the users to intuitively and quickly achieve the personalized design of 3D models by editing and modifying a few semantic parameters, so that the conceptual design of the product can be popularized.

In recent years, 3D shape retrieval is a fundamental issue in computer vision and pattern recognition, informative and discriminative features improve the retrieval accuracy [1][1][2], but many features are of high dimension that results in computational complexity [3]. Xie J et al. [2] proposed a deep shape descriptor based on stacked local convolutional auto-encoder [4] and deep belief network [5], which was insensitive to geometric structure variations, although the shape descriptors of different models were distinctive, the model had small variations within the class. Liu et al. [6] and Lun et al. [7] utilized a set of geometric feature descriptors and meaningful segmentations of shapes to compare 3D models, but it was hard to find a set of explicit descriptors to represent the style of shapes well. Lim I et al. [8] introduced an approach for style similarity learning of 3D shapes using deep metric learning that didn't depend on pre-computed descriptors, and style identification performance will be better in the setting of comparably smaller 3D shape collections. Li Y et al. [9] presented a method for sketch-based 3D model retrieval, which used a composite feature combining global and local feature of 2D views. It was especially suitable for sketches with rich details, however it would take more time during retrieval because a variety of sketch styles of different 3D models needs to be handled.

On the other hand, domestic and foreign scholars have conducted relevant research on the 3D model deformation technology and obtained certain results. Yumer et al. [10] defined the deformation handles to allow the user to freely select the deformation direction of the 3D model. Then, a set of semantic attributes [11] were used to create geometric deformation, while too many kinds of semantic attributes may cause the failure of deformation mode; Kunjin He et al. [12] presented a hierarchical framework for free-form surface feature parameters, it supported the user to intuitively edit and modify surface features, but lacked constraints between the framework levels. Then on the basis of them, they [13] focused on a reconstruction and characterization method for the local surface of models, it can edit and modify local surfaces, while the constraint line cannot be added automatically; Demisse et al. [14] applied the deformation of n dimensional shape space, although it has good robustness in shifting and overlapping on local shape, the computational cost is higher; Peizhong Liu et al. [15] introduced a model reconstruction method based on landmark depth estimation and shape deformation to achieve a more realistic 3D model, however, the details of the 3D model will be weakened with the increasing accuracy of 3D landmark estimation. Design optimization is a key component in the product development process of automotive industry, aircraft construction [16][17], and naval architecture. The complex product design process requires concurrent optimization methods to deal with different requirements in deformation [18][19]. Dai G et al. [20] extracted a data-driven shape descriptor to solve the problem of noise in 3D model deformation, and this method also has a better robustness when the degree of deformation is larger.

Although customized products have been open to the market, mass production or customized production usually have shortcomings, such as the time-consuming design process, high price, lack of high-level semantic parameters and disability of the real-time display of products appearance. Therefore, based on each component of the model, the system sets different functions for various deformation handles, and also sets the internal and external constraints for the components. It provides the user with a customized operation interface, which is convenient for users to directly edit and modify the 3D information of the deformation handle through high-level semantic parameters, so as to realize the hierarchical deformation of the model. Taking vehicles as 3D models, our method mainly includes: i) the reconstruction of the specific 3D model, ii) the replacement of the free-form surfaces based on each component of the 3D model, iii) the function setting for various deformation handles, iv) the calculation of constraints during deformation.

2 PARAMETRIC DEFORMATION FRAME OF 3D MODEL

The deformation process of 3D models based on semantic parameters generally includes reconstructing the 3D model, replacing the free-form surfaces of the model, setting deformation handles, calculating internal and external constraints, as well as parameterizing deformation handles, the flow is shown in Fig. 1.

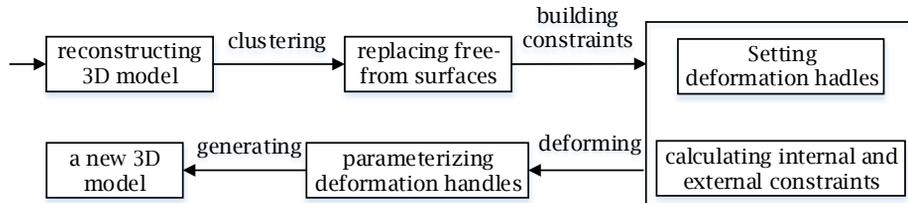


Figure 1: Parametric deformation of 3D model.

According to the above designing process, the parametric deformation steps of the 3D model are as follows:

Step 1. Loading the existing 3D complex models, grouping model surfaces into different components and reconstructing the 3D model;

Step 2. Clustering surfaces of each component, and replacing the free-form surfaces with the quadric surfaces;

Step 3. Establishing constraints, including setting deformation handles, internal and external constraints of the components;

Step 4. Parameterizing the deformation handles to realize the deformation of the 3D model.

Among them, Step 2 and Step 3 are the main contents. Generate the simplified 3D model composed of quadric surfaces through the above processes, then, Set different types of deformation handles on the corresponding components, which is mentioned in section 5.1. In order to effectively standardize the structural deformation of the 3D model, building the internal and external constraints of each component is necessary. Then, editing the semantic parameters can motivate the customized deformation of the 3D model, which means modifying the X-axis, Y-axis and Z-axis coordinate value of the deformation handle.

3 3D MODEL RECONSTRUCTING

A method of rebuilding 3D complex models is proposed based on the feature attributes of components. Based on the user-specified component, this method searches for the most similar component from a set of components and aligns automatically to form a new model. The core of the method lies in that the system automatically finds the new component that best matches with the user-specified refactoring component. First, the feature descriptor of each component is set to mark the properties of the component, where, the descriptor is composed of component surfaces' average normal vector n and the principal component vectors e_1, e_2, e_3 ; Second, the similarity of different descriptors is calculated with Euclidean distance.

Taking five 3D models of scooters as examples, like Figure. 2(a), any model (like a white scooter) is selected to be reconstructed as T. The five 3D models are composed of different components, like the yellow scooter, it is divided into a1, a2, a3, a4, a5, a6 corresponding to the head, the seat, the body, the front wheel, the rear wheel, and the kickstand respectively, and the number of all components of the five models is 28. Firstly, specify the head a1 of white scooter T as the component to be reassembled, calculate eigenvectors of each component's descriptor. Then, calculate the Euclidean distances of descriptors between all components and the head a1, which are listed in Tab. 1. The component a1' that is the yellow head has the smallest Euclidean distance and is replaces with a1. Finally, ensure a1' automatically aligned with T, then, a new 3d

complex model T' is formed. In brief, as shown in Figure. 2(b), the white scooter composing of the yellow head, the green body, the green front wheel, the red seat and the red front wheel, can be reconstructed with the above methods.

Types	Head	Seat	Body	Front Wheel	Rear Wheel	Kickstand
Yellow Scooter	2.4130	2.7457	2.4203	2.4897	3.5381	3.8416
Red Scooter	2.5192	2.6147	2.7599	3.2184	3.2183	3.2748
Blue Scooter	2.4859	2.6819	2.6749	2.4059	3.0917	3.1461
Green Scooter	2.6352	2.8932	2.9751	2.7761	2.7643	

Table 1: Euclidean distances of descriptors between all components and the head a1.

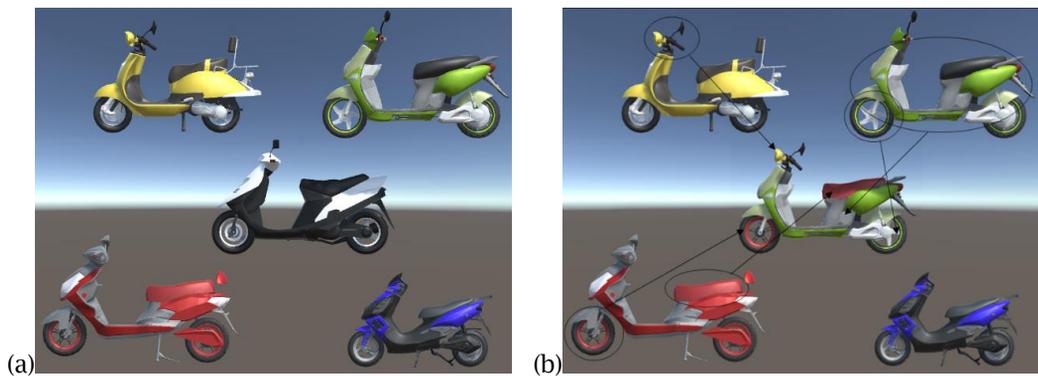


Figure 2: 3D reconstruction mode of scooters.

4 3D MODEL SIMPLIFYING

After simplifying and grouping the 3D model, the 3D model is divided into different components (like the white scooter is divided into head, seat, body, front and rear wheels). First of all, the sub-surfaces that make up each component are clustered, and then the sub-clusters are re-clustered with similar geometric information; then, according to the discriminant equation of quadric surfaces, like the plane, the sphere and the cylinder, the free-form surface is replaced with quadric surface.

4.1 Free-Form Surfaces Clustering

Each component is composed of several free-form surfaces, it is necessary to distinguish the types of the free-form surfaces by collecting the feature information. There two steps, one is clustering the surfaces with high similarity into a cluster by sub-surface clustering, and the other is re-clustering the surfaces with geometric feature information by mean-shift clustering.

4.1.1 Sub-Surfaces Clustering

The Hausdorff ranging method [21] was used to judge the similarity of the pairwise surfaces of each component in this paper. Hausdorff distance is a measure to describe the degree of similarity

between two sets of points, which refers to the maximum value of all distances from one set to the nearest point in another. Hausdorff ranging method is used to respectively compare the surface similarity of each components. Calculate the Hausdorff distance between any two surfaces A and B selected in a component and compare with the set threshold D . When Hausdorff distance is less than D , it is considered that the surface A and B are similar and can be classified into the same cluster. The mathematical description of the Hausdorff distance is defined as:

$$\begin{aligned}
 H(A, B) &= \max(h(A, B), h(B, A)) \\
 h(A, B) &= \max_{a \in A} \min_{b \in B} \| a - b \| \\
 h(B, A) &= \max_{b \in B} \min_{a \in A} \| b - a \|
 \end{aligned}
 \tag{1}$$

Among them, a and b are a vertex location information of surface A and surface B respectively. The method of defining threshold D in this paper is:

$$D = (\max_{B_i \in S} h(A, B_i) + \min_{B_i \in S} h(A, B_i) + \max_{B_i \in S} h(B_i, A) + \min_{B_i \in S} h(B_i, A)) / 4
 \tag{2}$$

Among them S is the set of surfaces to be classified, A is a specified surface, B_i is an arbitrary surface in the set.

After the first clustering of the sub-surface composing each component (e. g., body), several different sub-surface clusters $c = \{c_1, c_2, c_3 \dots c_m\}$ are obtained, then the characteristic information $x = \{e_1, e_2, e_3, n, k\}$ of each sub-surface in the c_m is calculated. e_1, e_2, e_3 are the principal component vectors describing the distribution information of a sub-surface vertex; n is a normal vector describing a sub-surface; $k = \{k_1, k_2, k_3\}$ are the Gauss curvatures describing a vertex j , which is calculated according to the Gauss-Bonnet theorem:

$$\iint_A k dAr = 2\pi - \sum_j \theta_j
 \tag{3.1}$$

$$k = 2(k - k_{min}) / (k_{max} - k_{min}) - 1
 \tag{3.2}$$

Among which, Ar refers to the sum of areas of all triangular faces related to a vertex j and θ_j refers to the sum of angles that relates to a vertex j . Then, the Gauss curvatures of vertexes located on a sub-surface are normalized to $[-1, 1]$, where k_{max} is the maximum Gauss curvature of all vertices in the surface, k_{min} is the minimum.

Then, the Gauss curvatures of all the vertices from a sub-surface are taken as the statistical data, the number of groups is 10. With the distribution of statistical data, find the top three frequency groups and calculate the mean value of each group that is respectively k_1, k_2 and k_3 . Taking the white head component which contains 15 free-form surfaces as an example, 3 different sub-clusters $c = \{c_1, c_2, c_3\}$ are obtained with Hausdorff distance method, c_1, c_2 , and c_3 contains 4, 5, and 6 free-form surfaces respectively. Then, the characteristic information of each free-form surface in sub-cluster c_1 is shown in Tab. 2, where all probabilities remain exact to two digits after the decimal point.

e_1, e_2, e_3	n	k_1, k_2, k_3
-0.52, 0.17, 0.84		
0.80, 0.27, 0.54	-0.14, -0.99, 0.00	0.12, 0.88, 0.72
0.32, -0.95, 0.00		
0.42, -0.38, 0.83		
-0.57, 0.60, 0.56	0.87, 0.50, 0.02	0.00, 0.00, 0.18
0.71, 0.71, 0.031		

0.28,-0.40,-0.87		
-0.96,0.13,-0.23	0.26,-0.97,0.00	0.20,0.25,0.48
-0.02,0.91 -0.42		
0.00,-1.00,-0.00		
1.00,0.00,0.00	-0.87,0.50,0.00	0.16,0.52,0.23
0.00,0.00,-1.00		

Table 2: Characteristic information of each free-form surfaces in sub-cluster c_1 .

4.1.2 Mean-Shift Clustering

The second clustering of different components of the 3D model is successively carried out, first, the sub-clusters $c=\{C_1,C_2,C_3...C_m\}$ are regarded as the clustering input; then, according to the Mean-Shift Algorithm [22], the clusters $C=\{C_1,C_2,C_3...C_n\}$ of sub-surfaces are calculated. The mathematical description of the mean-shift vector is defined as:

$$m(x) = \frac{\sum_{x_i \in N(x)} e^{-\|x_i - x\|} \cdot x_i}{\sum_{x_i \in N(x)} e^{-\|x_i - x\|}}$$

(4)

Among which, $m(x)$ is a mean-shift vector, x is the 15-dimensional eigenvector of a sub-surface, $N(x)$ is the neighbor of x and x_i is an arbitrary vector in $N(x)$. When $m(x)$ converges to a very small value, and if the distance between the central value of the present $N(x)$ and the central value of another existing cluster is small enough, then the two clusters are clustered together, otherwise, a new cluster is generated. The white head clusters $C=\{C_1,C_2,C_3\}$ can be obtained by applying the above algorithm to Sub-clusters c , which respectively contains 5,5,5 free-from surfaces.

4.2 Quadric Surface Replacement

The surface clusters $C=\{C_1,C_2,C_3...C_n\}$ from the component (such as the body) of a 3D model are generated after free-from surfaces clustering, replacing the free-from surfaces included in the surfaces cluster with the quadric surfaces. The result is beneficial for setting the deformation handles with semantic parameters on the basis of the feature of the quadric surface. The possibility of a certain surface cluster C_n is calculated with the discriminant [11], it can tell whether the possibility of C_n belongs to planar probability P_p , spherical surface probability P_s , or cylindrical surface probability P_c :

$$P_p = \sum_i^n \frac{C_i^p}{n}, P_s = \sum_i^n \frac{C_i^s}{n}, P_c = \sum_i^n \frac{n_c}{n}$$

$$C_i^p = \frac{2(\sigma_2 - \sigma_3)}{\sigma_1 + \sigma_2 + \sigma_3}, C_i^s = \frac{3\sigma_3}{\sigma_1 + \sigma_2 + \sigma_3} \quad (5)$$

Among them, n is the number of free-from surfaces contained in a cluster C_n , $\sigma_1, \sigma_2, \sigma_3, \sigma_1 > \sigma_2 > \sigma_3$ are the standard deviations of the point set derived from a surface i . n_c is the number of surfaces which has originated from a basic cylinder primitive [24]. If $P_p \geq 0.9$ or $P_s \geq 0.9$ or $P_c \geq 0.9$, then all free-from surfaces in C_n are characterized by the quadric surface with the highest probability calculated, and if $P_p < 0.9$ or $P_s < 0.9$ or $P_c < 0.9$, then all free-from surfaces in C_n maintain their original characteristics. Take the head component as an example, the standard deviation of the point set sampled from three surface clusters $C=\{C_1,C_2,C_3\}$ respectively

can be obtained, of which the surface cluster C_1 includes 5 free-from surfaces, then the standard deviation:

$$\sigma_1 = \{0.166, 0.154, 0.162, 0.185, 0.197\}, \sigma_2 = \{0.160, 0.149, 0.155, 0.180, 0.190\}, \sigma_3 = \{0.011, 0.015, 0.013, 0.000, 0.005\}$$

and $n_c = 0$ are obtained. The free-from surfaces in C_1 are replaced by the plane when $P_p=0.90$, $P_s=0.08$, $P_c=0.00$. Figure. 3 shows that the simplified 3D scooter model is divided into 5 different components that contain free-form surfaces replaced by the quadric surface.



Figure 3: Simplified 3D scooter model.

5 CONSTRAINTS BASED ON POINT CLOUD MODEL

When a component is chosen to be deformed, three types of deformation handles will be defined on the quadric surfaces of each component based on different functions. Then, the internal and external constraints are calculated to improve the rationality of 3D model deformation.

5.1 Classification of Deformation Handles

Different types of deformation handles are set on each component of the simplified 3D model. Deformation handles of each components are represented by different shapes (E.g., dots are used to indicate deformation handles on the body), as shown in Figure. 4, and divided into three types on the basis of different functions [11] in deforming

(1) The user-editing handle, which is controlled by the user to deform the 3D model, contains the defined semantic parameters. Its three-dimensional coordinate value automatically modified in accordance with user's demands on the 3D model appearance.

(2) The system-anchored handle, which is a fixed point and holds the model structure automatically, has global symmetry with the user-editing handle. It is used to reduce the number of fixed points marked by the user.

(3) The system-manipulated handle, which automatically realizes other components deformation, is determined by the user-editing handle and the system-anchored handle. The optimization equation to update the system-manipulated handle's position is shown below [11]:

$$\text{minimize } \sum_i \frac{1}{v_j} |m_j - s_j|, \text{ subject to } t_j = u_j, \forall t_j \in T \quad (6)$$

wherein, s_i is the value of the updated position, N is the number of the system-manipulated handles, and T is the component deformation types (X, Y, Z -axis translation, rotation and scaling); u_j is the deformation types specified by the user; t_j is one of them; m_i, v_i is respectively the mean

value and variance of vertices sampling from the regions controlled by system-manipulated handles.

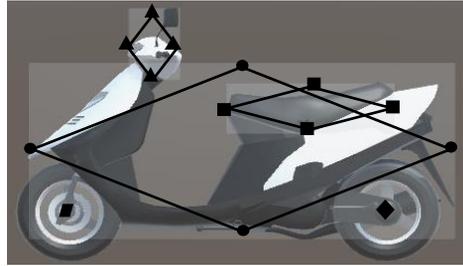


Figure 4: Different deformation handles of the simplified 3D scooter model.

5.2 Constraints Based on the Component to Be Deformed

In order to ensure the 3D model orderly deformation, constraints are set on different components, during the deformation design course from a component (such as the front wheel) to the whole (like the white scooter as a whole). When a certain component M is deformed, the internal constraints are defined to limit the variation of the component M . Moreover, the external constraints include the internal constraints of other components, and constraints between M and other components $MM=\{M_1, M_2, M_3..M_n\}$. Assuming the body M is chosen to be deformed, then $MM=\{M_1, M_2, M_3, M_4\}$ corresponds to the head, seat, front wheel and rear wheel components respectively. Figure. 5 shows the internal and external constraints of the body deformation, Figure. 5(a1) is the point cloud model of the scooter, and Figure. 5(a2) is a combination model of point cloud and free-form surfaces. The example diagram shown in a curved surface model carries convenience for marking variables.

5.2.1 Internal Constraints of component to Be Deformed

Figure. 5(b) shows the internal constraints of the body. According to the body center, the influence regions of *Area1* and *Area2* controlled by the user-editing handle e and the system-anchored handle a are divided. *The new position p_{new} of p_i is determined by the relationship between the influence factor k and the distance $D_{(e, p_i)}$. Constraints based on the point e and p_i are divided into three types, with regard to the distance $D_{(e, p_i)} = \|e - p_i\|$: (1) when $D_{(e, p_i)} < 0.1$, then $k=1$, $p_{new} = p_i$; (2) when $0.1 < D_{(e, p_i)} < 1$, find out the minimum distance D_{min} and maximum distance D_{max} within this range, then $k = 1 / (D_{min} + D_{max})$, $p_{new} = p_i + temp \cdot k$; (3) when $D_{(e, p_i)} > 1$, then $k = 1 / D_{(e, p_i)}$, $p_{new} = p_i + temp \cdot k$. In the same way, the new position of a vertex in *Area2* affected by point a can also be obtained.*

5.2.2 External Constraints of Component to Be Deformed

Suppose the body length is increasing, calculate the constraints between deformation handles $p=\{e, a, E\}$, $E=\{E_1, E_2, E_3, E_4\}$, that is, the internal constraints of other components. The deformation handles on different components are shown in Figure. 5(c), where point e on the body is the user-editing handle, point a is the system-anchored handle, and E is the set of system-manipulated handles located on other components. $E_1=\{e_{11}, e_{12}\}$, $E_2=\{e_{21}, e_{22}\}$, $E_3=\{e_3\}$, $E_4=\{e_4\}$ are respectively correspond to the head, the seat, the front wheel and the rear. The changing value of $E=\{e_{11}, e_{12}, e_{21}, e_{22}, e_3, e_4\}$ can be obtained with Equation. (6). Taking the seat as an example, the influence regions are divided into *Area21* and *Area22* which are controlled by the system-manipulated handles e_{21} and e_{22} . The vertices effected by point e_{21} and e_{22} in different influence regions drive the seat deformation on the basis of the similar internal constraints, which are mentioned in section 5.2.1. In brief, other components will deform like the deformed body component.

The establishment of constraints between adjacent components can effectively reduce the phenomena of overlap and suspension during deforming. The steps of setting constraints between adjacent components include: (1) Find the influence region $EA = \{EA_1, EA_2, EA_3 \dots EA_n\}$ between M and adjacent components $M_1, M_2, M_3 \dots M_n$ corresponding to $EA_1, EA_2, EA_3 \dots EA_n$. (2) Define constraints between the arbitrary vertex p_i on the influence region EA_n and the deformation handle located on M ; (3) Get the new position of point p_i on the influence region EA_n on the basis of the constraints defined in step(2).

Take the seat M_2 influenced by the body M as an example. First, the influence region EA_2 is obtained through the 3D overlapping space between M and M_2 ; Secondly, the constraints between deformation handles e, a and point p_i located on the influence region EA_2 are respectively obtained as mentioned in section 4.2.1. Constraints between the body M respectively with the head M_1 , front wheel M_3 and rear wheel M_4 are obtained in the same way. Figure. 5(d) shows the scooter deformation when its body length is increased by $temp$.

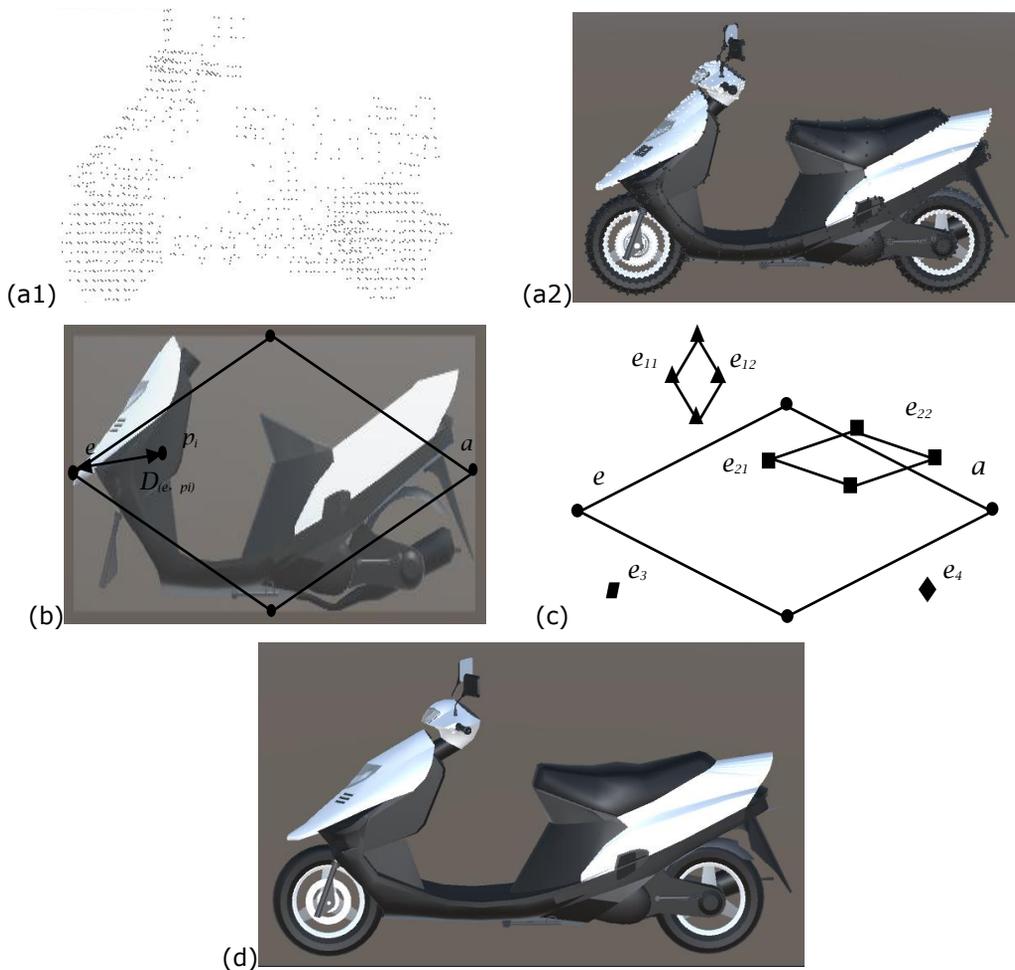


Figure 5: Internal and external constraints of the body deformation: (a) Point cloud model of the scooter, (a2) Combination model of the point cloud model and the surface model, (b) Internal constraints of the body, (c) Deformation handles located on five components, (d) White scooter deformation when it's body length is increased by $temp$.

6 IMPLEMENTATION

The methods proposed are tested on VS2010 and Unity 5.5.3 platforms. Experiments are designed to illustrate 3D model deformation with self-defined parameters. Examples taken include the white scooter model, the reconstruction of scooter model, and the bicycle model.

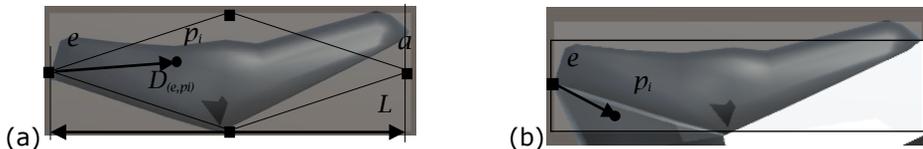
6.1 Experiment1: A White Scooter Model

Figure. 6 shows the parametric deformation of the white car model. The experiment is carried out with Individual and composite components. After using methods mentioned in section4, the simplified model (Figure. 3) of the five components can be obtained.

The individual component (the seat component) is tested firstly. The user-editing handle e and the system-anchored handle a are set on the seat component, as shown in Figure. 6(a). $D_{(e,p_i)}$ is the distance from point e to p_i and L is the initial length of the seat.

Firstly, the seat center is used to divide influence region of the point e and point a into two parts. According to the internal constraint condition mentioned in section 5.2.1, the influence region controlled by point e change with the distance $D_{(e,p_i)}$, and the other components controlled by point a deforms in the same way. Secondly, calculate the constraints of the other component, like the head, the body. The updated position of system-manipulated handles located on other components is gotten with the Equation. (6), then, each component can also be deformed by using the method in section 5.2.1. Thirdly, to make sure that the adjacent components are aligned in the deformation process, set the external constraints to restrict the change of vertices. In terms of this test, these vertexes are located on the body, and also in the seat component's 3D space that is marked in a black box, as shown in Figure. 6(b). Then, these vertexes in a black box are changed respectively with the different distances between point e and point p_i . Lastly, modify the X-axis coordinate value of point e to achieve the deformation of the white scooter model when the seat length is increased, as shown in Figure. 6(e).

In addition, the other individual component (the front wheel component) is verified as follow. Firstly, set the user-editing handle e on the center of front wheel component, as shown in Figure. 6(c). \mathbf{D} is the vector from point e to p_i which is any vertex in the front wheel, and R is the initial radius of this component. Secondly, increase the radius of the front wheel by $p_{new}=p_i + \mathbf{D}*R1$. Thirdly, deform other components with Equation. (6). At the same time, vertices, which are chosen in the same way as the third point in the previous paragraph, are restricted by the user-editing handle e , as shown in Figure. 6(c). Finally, ensure that the lowest point of the front wheel is static and increase the X-axis coordinate value of point e , so that the front wheel components are enlarged and other components are changed, as shown in Figure. 6(f). Figure .6(h) is the result of increasing the Y-axis coordinate values of point e located on the front wheel, then, increasing the X-axis coordinate value of the point e when it is located on the seat, as well as increasing the X-axis coordinate value of the point e when it is located on the body.



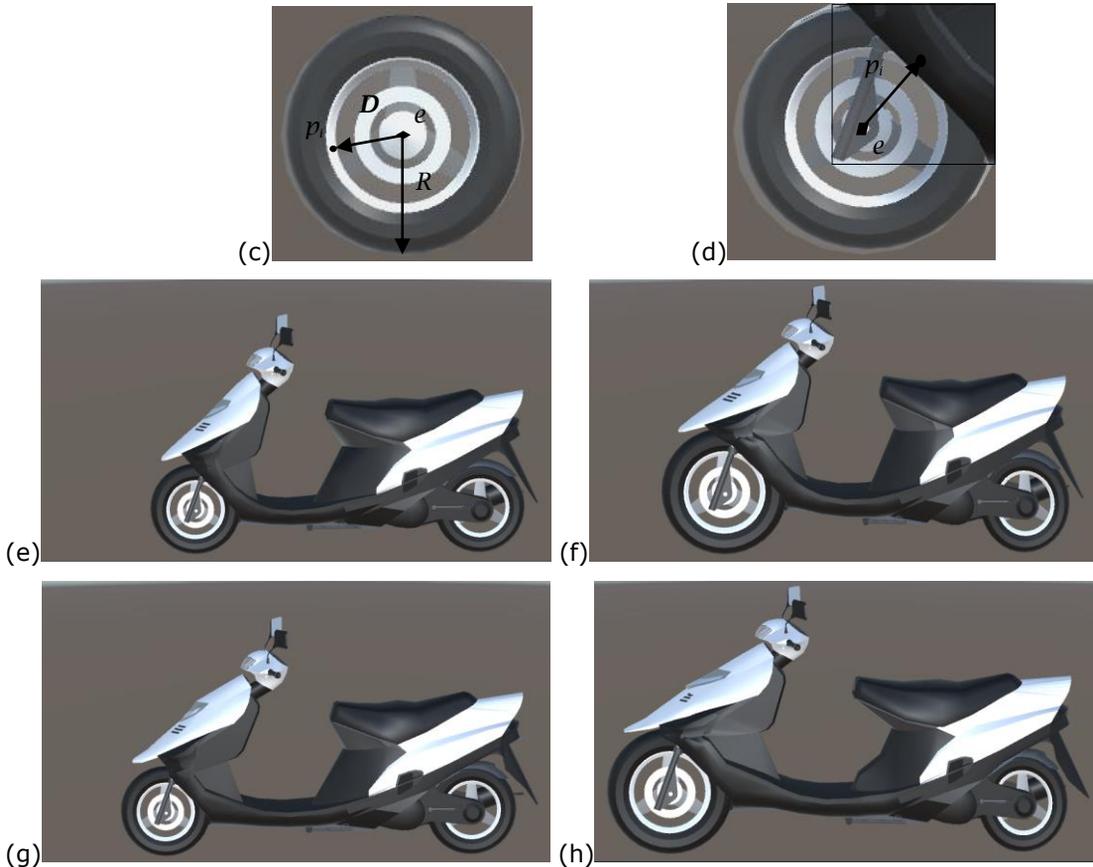


Figure 6: Parametric deformation of the white scooter model: (a) Seat component when length is L , (b) External constraints between the seat and the body, (c) Front wheel component when radius is R , (d) External constraints between the front wheel and the body, (e) White scooter model deformation when the seat length is increased by 11%, (f) White scooter model deformation when the front wheel radius is increased by 34%, (g) White scooter model deformation when the body length is increased by 15% as described in section 4, (h) Combined deformation of the above three components e, f and g.

6.2 Experiment2: A reconstruction of Scooter model

On the other hand, Figure. 7 shows the parametric deformation of 3D reconstruction model obtained from section 2. In order to improve deformation efficiency, the number of the model vertices are simplified from 39719 to 22288, it may result in some rough surfaces as shown in Figure. 7(a). During the surface clustering, statistical data like normal vector or Gauss curvatures still are obtained by vertices located on the original model. Then, set the deformation handles on the appropriate components. Suppose to increase the length of body component, then point e and point a are correspond to user-editing handle and the system-anchored handle, the other points are system-manipulated handles, as shown in Figure. 7(b). Finally, after setting constraints as mentioned in Section 5.2, modify the coordinate value of any axis of point e according to the user's demands. Figure. 7(c) shows the body deformation, it means the X-axis coordinate value of the user-editing handle is changed, and Figure. 7(d) shows the increase of rear wheel radius when increasing the Y-axis coordinate value of point e located on the rear wheel component.

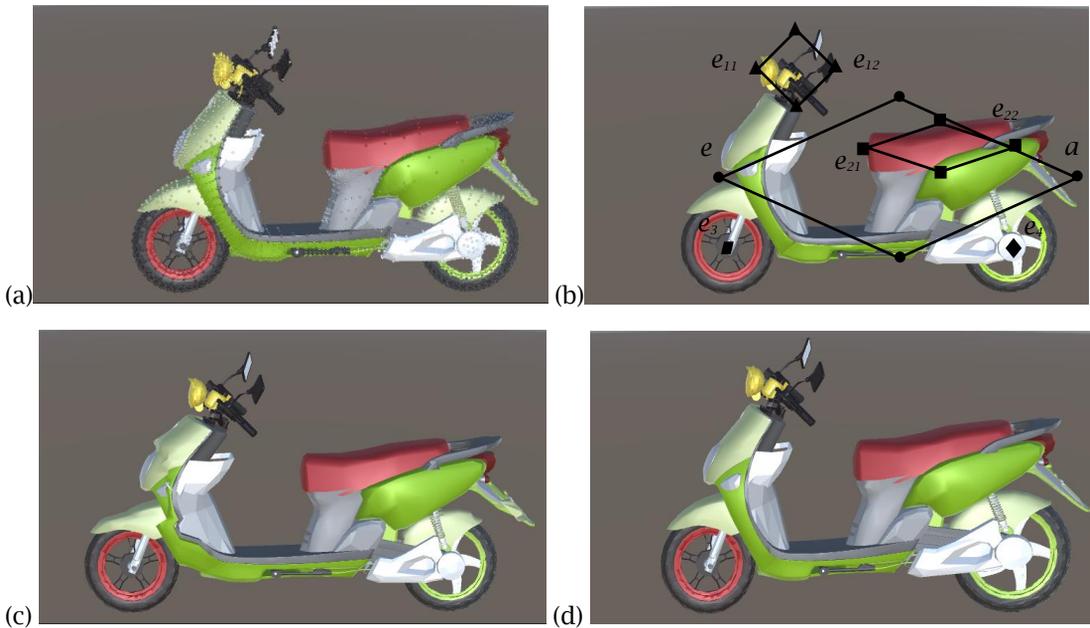


Figure 7: Parametric deformation of the 3D reconstruction model: (a) Combination cloud model of 3D reconstruction model, (b) Deformation handles located on five components, (c) D reconstruction model deformation when the body length is increased by 12%, (d) 3D reconstruction model deformation when the rear wheel radius is increased by 21%.

6.3 Experiment3: A Bicycle Model

To test the more applicability of above methods, a 3D bicycle model is taken as example. Figure. 8 shows the parametric deformation of the bicycle model from two aspects: deforming a component separately and deforming multiple components simultaneously.

The bicycle model are divided into seven components. There are two steps to deform the bicycle model. The first step is replacing the free-form surface by quadric surface, according to Section 4. Take the basket component as an example. Firstly, two different sub-clusters $c = \{c_1, c_2\}$ of the basket are obtained with Hausdorff distance method. Four and twelve free-form surfaces in sub-cluster c_2 is listed in Tab. 3. Secondly, standard deviations $\sigma_1, \sigma_2, \sigma_3$ of vertices are calculated,

$$\sigma_1 = \{31.6447, 31.3926, 33.2536, 32.3364\}, \sigma_2 = \{30.4717, 26.7712, 31.4270, 28.1161\}, \sigma_3 = \{0.3242, 0.3316, 0.3101, 0.2887\}$$

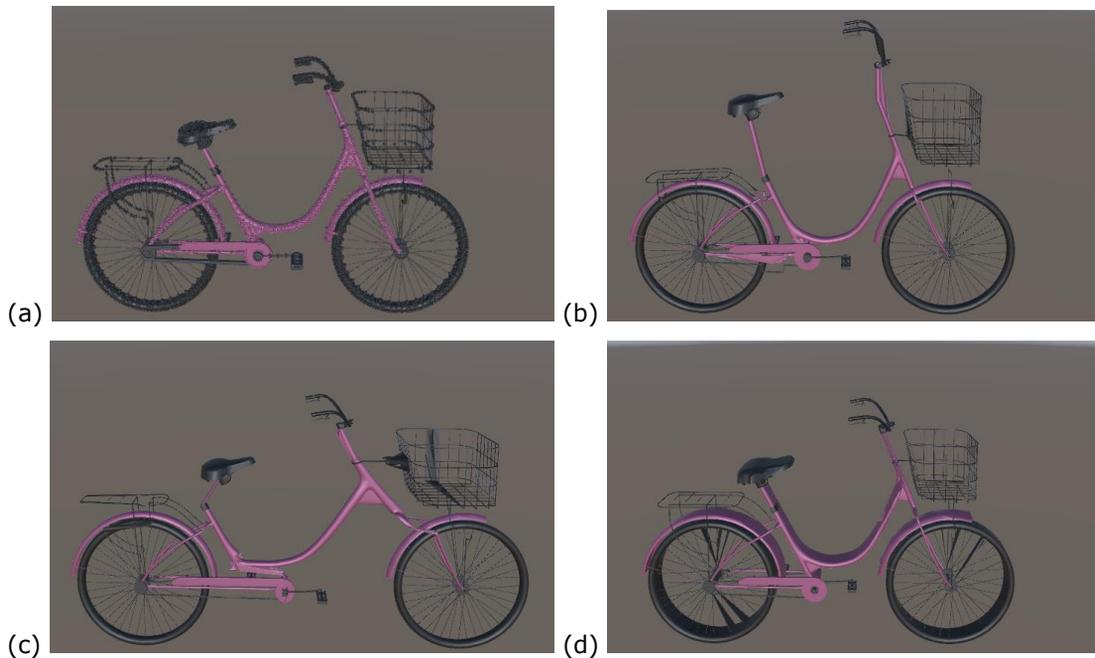
then, the probability $P_p=0.94, P_s=0.02, P_c=0.00$ can be calculated by using the Equation. (5), which is mentioned in Section 4.2.

e_1, e_2, e_3	n	k_1, k_2, k_3
-0.54, 0.84, 0.00		
0.84, 0.54, 0.00	0.92, 0.34, 0.00	0.30, 0.11, 0.48
-0.00, -0.00, 1.00		

-0.11,-0.99,-0.00		
-0.99,0.11,0.00	0.41,0.41,0.81	0.16,0.28,0.38
0.00,-0.00,1.00		
-0.57,-0.82,-0.00		
0.82,0.57,0.00	-0.89,0.53,0.04	0.36,0.17,0.28
0.00,-0.00,1.00		
0.24,-0.97,0.00		
-0.97,0.24,-0.00	-0.09,-0.46,-0.89	0.01,0.12,0.20
-0.00,0.00,1.00		

Table 3: Characteristic information of partial free surfaces in sub-cluster c_2 .

The second step is setting deformation handles, and establishing internal and external constraints based on Section 5. The last step is modifying the parametric of user-editing handle e under different circumstances. Figure. 8(b)-(d) are the results of increasing respectively the Y-axis, X-axis and Z-axis coordinate values of the point e located on the body Figure. 8(e) is the result of decreasing the Y-axis coordinate value of the point e located on the front wheel, as well as increasing X-axis coordinate value of the point e when it is located on the body. Figure. 8(f) is the result of increasing the Y-axis and Z-axis coordinate values of point e located on the body, as well as increasing the Y-axis coordinate value of the point e when it is located on the front wheel.



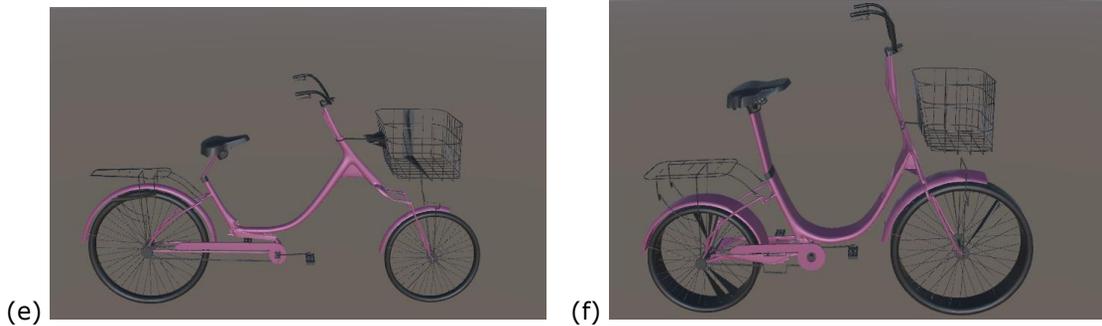


Figure 8: Parametric deformation of the bicycle model: (a) Point cloud model of bicycle model, (b) The bicycle deformation when the body height is increased by 27%, (c) The bicycle deformation when the body length is increased by 31%, (d) The bicycle deformation when the seat width is increased by 67%, (e) Combined deformation when the body length is increased by 31% and the front wheel radius is decreased by 33%, (f) Combined deformation with the above two components B and D, and meanwhile the rear wheel radius is decreased by 33%.

7 CONCLUSION

Upon the existing 3D models, a method of 3D complex model reconstruction and deformation based on semantic parameters is proposed. Based on the user-specified component, this method searches for the most similar component in a database and aligns automatically to form a new model, greatly enriching the test model types. Two main problems are solved during deforming. Replace free-form surfaces of the 3D model with the regular quadric surface, it can make the free-form surfaces deform easily. Establish constraints between the components of the 3D model, like the constraints between deformation handles and components. It can realize the orderly and hierarchical deformation of the 3D model. Although the main body of the constraint equations for different models is the same, the coefficients differ slightly. And the more imprecise the constraint sets, the worse the deformation is. These will be further studied and perfected in our future work.

Kujin Hei, <https://orcid.org/0000-0002-0843-858X>
Xia Sun, <https://orcid.org/0000-0003-3216-1418>

REFERENCES

- [1] Fang, Y.; Xie, J.; Dai, G.; Wang, M.; Zhu, F.; Xu, T.; Wong, E.: 3D deep shape descriptor, *Computer Vision and Pattern Recognition, IEEE*, 2015, 2319-2328.
- [2] Litman, R.; Bronstein, A.; Bronstein, M.; Castellani, U.: Supervised learning of bag-of-features shape descriptors using sparse coding, *Computer Graphics Forum*, 33(5), 2015, 127-136. <https://doi.org/10.1111/cgf.12438>
- [3] Xie, J.; Fang, Y.; Zhu, F.; Wong, E.: Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(7), 2015, 1335-1345. <https://doi.org/10.1109/TPAMI.2016.2596722>
- [4] Li, B.; Lu, Y.; Li, C.; Godil, A.; et al.: A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries, *Computer Vision & Image Understanding*, 131(C), 2015, 1-27. <https://doi.org/10.1016/j.cviu.2014.10.006>
- [5] Leng, B.; Guo, S.; Zhang, X.; Xionf, Z.: 3D object retrieval with stacked local convolutional autoencoder, *Signal Processing*, 112(C), 2015, 119-128. <https://doi.org/10.1016/j.sigpro.2014.09.005>

- [6] Rosin, P. L.; Sun, X.; Martin, R. R. et al.: Shape retrieval of non-rigid 3D human models, Eurographics Workshop on 3d Object Retrieval, Eurographics Association, 2014, 101-110.
- [7] Li, W.; Liu, T.; Hertzmann, A.; Funkhouser, T.: Style compatibility for 3D furniture models, ACM Transactions on Graphics, 34(4), 2015, Article No. 85. <https://doi.org/10.1145/2766898>
- [8] Lun, Z.; Kalogerakis, E.; Sheffer, A.: Elements of style: learning perceptual shape style similarity, ACM Transaction on Graphics, 34(4), 2015, Article No. 84. <https://doi.org/10.1145/2766929>
- [9] Lim, I.; Gehre, A.; Kobbelt, L.: Identifying Style of 3D Shapes using Deep Metric Learning, Symposium on Geometry Processing, Eurographics Association, 2016, 207-215. <https://doi.org/10.1111/cgf.12977>
- [10] Li, Y.; Lei, H.; Lin, S.; Luo, G.: A new sketch-based 3D model retrieval method by using composite features, Multimedia Tools & Applications, 77(11), 2017, 1-24.
- [11] Yumer, M. E.; Kara, L. B.: Co-constrained handles for deformation in shape collections, ACM Transactions on Graphics, 33(6), 2014, 1-11. <https://doi.org/10.1145/2661229.2661234>
- [12] Yumer, M. E.; Chaudhuri, S.; Hodgins, J. K.; Kara, L. B.: Semantic shape editing using deformation handles, Association for Computing Machinery Transactions on Graphics, 34(4), 2015, 86. <https://doi.org/10.1145/2766908>
- [13] He, K.; Zhao, Z.; Geng, W.; Chen, Z.; Zhu, Y.: Parametric representation and implementation of freeform surface feature based on layered parameters, Journal of Computer-Aided Design and Computer Graphics, 26(5), 2014, 826-834.
- [14] He, K.; Wang, L.; Chen, Z.; Zhao, Z.: Reconstruction and featurization of local region based on CAD surface models, Computer Integrated Manufacturing System, 20(10), 2014, 2360-2366.
- [15] Demisse, G. G.; Aouada, D.; Ottersten, B.: Deformation Based Curved Shape Representation, IEEE Transactions on Pattern Analysis & Machine Intelligence, PP(99), 2017, 1-1.
- [16] Liu, P.; Hong, M.; Wang, M.; Gu, P.; Huang, D.: 3D face reconstruction via landmark depth estimation and shape deformation, Multimedia Tools & Applications, 76(2), 2017, 2749-2767. <https://doi.org/10.1007/s11042-016-3259-8>
- [17] Sieger, D.; Menzel, S.; Botsch, M.: On Shape Deformation Techniques for Simulation-Based Design Optimization, Sema Simai Springer, 5, 2015, 281-303. https://doi.org/10.1007/978-3-319-06053-8_14
- [18] Sieger, D.; Gaulik, S.; Achenbach, J.; Menzel, S.; Botsch, M.: Constrained space deformation techniques for design optimization, Computer-Aided Design, 72, 2015, 40-51. <https://doi.org/10.1016/j.cad.2015.07.004>
- [19] Richter, A.; Achenbach, J.; Menzel, S.; Botsch, M.: Multi-objective Representation Setups for Deformation-Based Design Optimization, Proc. EMO 2017, 10173, 2017, 514-528. https://doi.org/10.1007/978-3-319-54157-0_35
- [20] Alizadehkhameh, M. A.; Eshagh, M.; Jensen, A. B.: Optimization of deformation monitoring networks using finite element strain analysis, Journal of Applied Geodesy, 12(2), 2018, 187-197. <https://doi.org/10.1515/jag-2017-0040>
- [21] Dai, G.; Xie, J.; Zhu, F.; Fang, Y.: Learning a discriminative deformation-invariant 3D shape descriptor via many-to-one encoder, Pattern Recognition Letters, 83, 2016, 330-338. <https://doi.org/10.1016/j.patrec.2016.04.005>
- [22] Kim, Y. J.; Oh, Y. T.; Yoon, S. H.; Kim, M. S.; Elber, G.; et al.: Efficient Hausdorff Distance computation for freeform geometric models in close proximity, Computer-Aided Design, 45(2), 2013, 270-276. <https://doi.org/10.1016/j.cad.2012.10.010>
- [23] Carreiraperpiñán, M. Á.: A review of mean-shift algorithms for clustering, Computer Science, 2015
- [24] Yumer M. E.; Kara, L. B.: Co-abstraction of shape collections, ACM Transactions on Graphics, 31(6), 2012, 1-11. <https://doi.org/10.1145/2366145.2366185>