IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 March 2023, accepted 2 April 2023, date of publication 6 April 2023, date of current version 18 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265175

METHODS

A Pose-Normalization Method for Casting Voxel Models Using Second-Order Central Moment Matrix

SHUREN GUO, XUANPU DONG, DONG XIANG, AND HUATANG CAO

State Key Laboratory of Materials Processing and Die & Mould Technology, School of Materials Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Xuanpu Dong (dongxp@hust.edu.cn)

ABSTRACT Within the framework of casting process design, the efficient retrieval of three-dimensional (3D) computer-aided design (CAD) models could result in significant time and cost savings. However, this technique still suffers from inefficiency and inaccuracy because of the wide variety of casting models' poses. In this study, a method for normalizing the poses of casting models is proposed. This method constructs the transformation matrix through the eigendecomposition of the second-order central moment matrix calculated from the voxel casting model. Then the transformation matrix is applied to the casting model to get a normalized pose. An assessment approach for pose normalization is also suggested in the study, which measures the distance between poses normalized based on multiple poses of the same model. The study demonstrates that the pose-normalization approach reliably transforms distinct poses of the same model into a unified pose. The mean distance between normalized poses is 0.016 and the minimum distance is 0.010. The method's effects improve when the voxel size reduces.

INDEX TERMS Casting, voxel model, pose normalization, second-order moment, digital manufacturing.

I. INTRODUCTION

3D CAD models are the basic form of data for modern advanced manufacturing. The open CAD parts library TraceParts [1] consists of over 100 million CAD parts. It is expected that an increasing number of part models will be designed and stored shortly, and model retrieval will be an essential issue. In the foundry industry, the design of the casting process usually depends on past cases and experiences. The process design of a new part often refers to a proven process in the production practice, whether it is a part with a similar structure or a summarized experience model. It is often rather challenging to find existing casting and processes similar to the new part through text search.

Furthermore, model retrieval based on content is a more practical approach. Image retrieval based on content is a mature technology and has been applied in search engines, such as Google and Baidu, Inc. The fundamental method uses the features extracted from images as the index to build a repository of images and achieve retrieval [2]. The key feature of content-based retrieval is the unified pre-processing method of the input content and the extraction of content, see Figure 1.

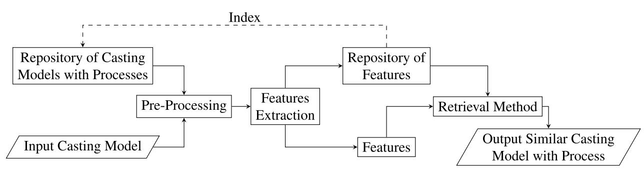


FIGURE 1. The diagram of the content-based retrieval method.

3D models have six degrees of freedom in translation and rotation. Whatever degree of freedom changes, it will become a new pose of the 3D model. Moreover, in computers the data representations of different poses of a model are different, and the features extracted from them may vary significantly.

There are two paths to avoid the interference of model transformation during model retrieval: (i) to achieve translation and rotation invariance in the feature extraction

The associate editor coordinating the review of this manuscript and approving it for publication was Wu-Shiung Feng.

algorithm; (ii) to perform a pose normalization of the model. In class (i), the invariance of transformation is achieved by the definition of the feature extraction algorithm itself, primarily by statistical methods to avoid interference with the model transformation, such as statistical shape description [3], frequency domain analysis [4], and deep learning methods [5], [6], [7]. By contrast, in class (ii), the invariance of transformation is achieved by normalizing the model's pose in the pre-processing. Compared to class (i), the latter can keep more partial structures. Meanwhile, deep-learning methods also use pose normalization as preprocessing [8]. Furthermore, with a normalized pose of the model, more features without transformation invariant could be considered to represent the model [9].

Besides, pose normalization is an essential pre-processing method. Principal Component Analysis (PCA) is a method for statistical analysis and simplification of data sets. It enables the projection of potentially correlated variables into a set of uncorrelated variables by orthogonal transformation. For example, Vilar et al. [10] used PCA method to normalize the objects' pose before computing the Histogram of Gradients (HOG) descriptor to recognize 3D volumetric objects in robotic vision systems. Kanaan and Behrad [11] also used the PCA method before extracting shape descriptors.

The study proposes a pose-normalization method for the voxel models, which constructs the transformation matrix from the current pose to a consistent pose by calculating the voxel model's second-order central moment matrix. It transforms the model with any pose to a normalized pose for further feature extraction, which provides an essential guarantee for the realization of automated structural analysis and decomposition of castings. This pose normalization is more than a pre-processing method for model retrieval or feature extraction. It provides a benchmark of models' poses, then a model's pose can be represented by a transformation matrix.

In the following sections, the study defines the voxel model and the second-order central moment matrix, and demonstrates the pose normalization method (Section III). Then the study proposes a method for assessing the pose normalization algorithm and investigates the effect of voxel size in the voxel model on the effect of pose normalization (Section IV).

II. RELATED WORKS

Vranic et al. [12] applied the PCA method to normalize the model with the set of vertices of triangular surface pieces as input to calculate the transformation matrix. However, it performs low compliance due to the uneven distribution of the triangles. Therefore, they assigned weights to each vertex according to the triangle area to solve this problem. Paquet et al. [13] used the set of centroids as input instead of vertices and also append weights. Vranic [14] proposed the Continuous PCA (CPCA) algorithm based on the original method, which calculated the central axis of the 3D model by integrating the points in each triangle. Papadakis et al. [15] used the normal vectors of triangles as input to calculate the central axis, which was named Normals PCA (NPCA). Chaouch and Verroust-Blondet [16] improved the algorithm based on the symmetry of the objects. Moreover, for the 3D point cloud models, Guo et al. [17] applied the PCA method to the point cloud models and found that it may cause significant instability due to data loss and lead to a wrong result. Yu et al. [18] trained a multi-head neural network to select a standard pose from sym-space and rot-space constructed from PCA results.

Besides PCA methods, Kazhdan et al. [19] defined a reflection symmetry descriptor that measures the model's reflection symmetry to find the principal symmetry axes of the model. Sfikas et al. [20], [21] proposed pose-normalization methods based on the reflection symmetry descriptor. Yuan et al. [22] calculated the best rotation axis and angle for the whole model through a probabilistic approach. Fu et al. [23] trained a support vector machine(SVM) using features like polygon points, centroids, and convex packets. It combined symmetry with machine learning methods. Czerniawski et al. [24] implemented a method for discriminating planar features from point clouds by clustering them based on their upright orientation. With the progress of artificial neural networks, Sedaghat et al. [25] applied network structures to pose normalization.

Learning-based methods are usually able to achieve effective results thanks to the support of large amounts of data. Therefore, they are very dependent on datasets. For problems that have not been resolved or lack of dataset, learning-based methods may not be effective at all. For casting models, there is no dataset of standard poses of models. At the same time, the standard poses of casting CAD Models have not been subjectively defined, and it is difficult to construct the dataset manually. Thus, analysis-based or calculation-based methods are more suitable.

Both point clouds and polygon surface models contain only the surface of the models. They are not solid representations and cannot express internal features well. Therefore, this research aims to exploit the 3D voxel model to solve the above problem.

III. THEORY

This section defines the voxel model and the second-order central moment matrix, as well as demonstrates the pose normalization method.

A. VOXEL MODEL

3D model represents a physical body using a collection of points in 3D space, connected by various geometric entities. The voxel model uses a stack of discrete voxels to form the solid by sampling the points. It is widely used in medical analysis [26] and geographic mapping. Furthermore, modeling and analysis based on the voxel model in industrial engineering are also gradually developing [27]. However, voxel models take up a large amount of storage space, making them incapable of expressing high-resolution 3D data [28].

In the space $\mathbb{R}^3$, a function $V = F(x, y, z)$, x, y, $z \in \mathbb{R}$ is able to represent the continuous distribution in space. By sampling and quantization, the continuous distribution is discredited into a three-dimensional array expressed as $f(x, y, z)$, where (x, y, z) are discrete coordinates. Thus a voxel model is represented by the following equation.

$$\begin{aligned} t &= f(x, y, z),\ x, y, z \in \mathbb{N} \\ or\ t &= f(\mathbf{v}),\ \mathbf{v} \in \mathbb{N}^3, \end{aligned} \tag{1}$$

where t means the mass of the voxel. For example, $t = 0$ means the current voxel is empty.

Obviously, the voxel model has less accuracy than mesh and point cloud because of the rasterization. However, it represents the solid of the object rather than the surface, which is closer to reality. Besides, it is easier to modify the voxel model and apply volume-related calculations to the voxel model, which is necessary for further processing.

B. SECOND-ORDER MOMENTS MATRIX

In mathematics, the moments of a function are quantitative measures related to the shape of the function's graph. These reflect some features of the function and have their own means. Regarding the voxel model as a function that represents mass in space, then the first moment is the center of the mass, and the second moment is the rotational inertia.

Since Hu [29] proposed the invariant moments of images, moments have been widely used in pattern recognition and image analysis. Sadjadi et al. [30] proposed a method to calculate the geometric moments of 3D models using the set of vertices and constructed invariant moments.

1) SECOND-ORDER MOMENTS OF SOLID MODEL

For a 3D continuous model, the moments are defined as

$$M_{pqr} = \int\int\int x^p y^q z^r f(x, y, z) dxdydz. \tag{2}$$

M_{pqr} is called the $(p + q + r)$-order moment of the model.

Central moments are defined as

$$\mu_{pqr} = \int\int\int (x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r f(x, y, z) dxdydz, \tag{3}$$

where $\bar{x} = \frac{M_{100}}{M_{000}}$, $\bar{y} = \frac{M_{010}}{M_{000}}$, $\bar{z} = \frac{M_{001}}{M_{000}}$.

The second-order central moments include six types, μ_{200}, μ_{020}, μ_{002}, μ_{110}, μ_{101}, μ_{011}. The second-order central moment matrix, also named covariance matrix, is constructed as following by arranging the second-order central moments in a matrix according to their footnotes.

$$C = \begin{bmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{bmatrix}. \tag{4}$$

If $\mathbf{v} = [x, y, z]^T$ denotes the point in the model, and $\mathbf{m} = [\bar{x}, \bar{y}, \bar{z}]^T$ denotes the center of mass, the matrix is calculated as follows,

$$C = \int (\mathbf{v} - \mathbf{m})(\mathbf{v} - \mathbf{m})^T f(x, y, z) d\mathbf{v}. \tag{5}$$

If it is assumed that the mass is evenly distributed in the model, moments are only related to the geometry of the model. Thus these moments are named ''geometric moments''. Furthermore, there exist a variety of values calculated from these moments that can remain invariant in the model transformation. Therefore, these values are named ''invariant moments''. For example, the study found the eigenvalues of the second-order central moment matrix would not change in the model rotation, which is proved in section III-C. The above two types of values are both derived from the second-order central moments and show certain characteristics of the model.

2) SECOND-ORDER MOMENTS OF VOXEL MODEL

The above calculations and formulas all integrate into the continuous model. While the voxel model is sampled from the continuous model, the calculation formula needs to be modified.

We consider the voxel as a cube with side length l and uniformly distributed internal masses. The coordinates of the voxel are considered the coordinates of the geometric center of the voxel. Recall that

$$\mathbf{v} = \mathbf{V_i} + [x, y, z]^T x, y, z \in [-l/2, l/2], \tag{6}$$

where $\mathbf{V_i}$ is the voxel of the model, $\mathbf{v}$ represents the point of the voxel $\mathbf{V_i}$, the center of mass of the voxel model denoted $\mathbf{m}$ is calculated as follows:

$$\mathbf{m} = \frac{\sum_{i=1}^{N} l^3 \mathbf{V}_i f(\mathbf{V}_i)}{\sum_{i=1}^{N} l^3 f(\mathbf{V}_i)}. \tag{7}$$

The second-order central moment matrix of voxel model is calculated as

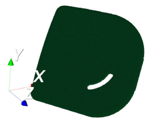
$$\begin{aligned} C &= \sum_{i=1}^{N} \int_{\mathbf{v} \in \mathbf{V_i}} (\mathbf{v} - \mathbf{m})(\mathbf{v} - \mathbf{m})^T f(\mathbf{v}) d\mathbf{v} \\ &=> \sum_{i=1}^{N} l^3 * ((\mathbf{V_i} - \mathbf{m})(\mathbf{V_i} - \mathbf{m})^T + \frac{l^2}{12} I) * f(\mathbf{V_i}), \end{aligned} \tag{8}$$

where N is the number of voxels.

C. POSE NORMALIZATION

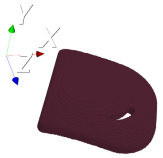
The study calculates the second-order central moment matrix of different poses of a model.

As shown in Figure 2, the eigenvalues of the second-order central moment matrix do not change during model rotation, but the matrix does. This may support that the matrix represents the rotation and eigenvalues represent something invariant. It is well known that the second-order central moment matrix can be eigen decomposed to a diagonal matrix formed by eigenvalues. This diagonal matrix tends to correspond to a



$$\begin{bmatrix} 2.1\times10^9 & 1.2\times10^8 & 8.8\times10^8 \\ 1.2\times10^8 & 2.7\times10^9 & -8.8\times10^7 \\ 8.8\times10^8 & -8.8\times10^7 & 4.2\times10^8 \end{bmatrix}$$
$$Eigenvalues = [2.8\times10^9, 2.4\times10^9, 2.9\times10^7]$$

(a)



$$\begin{bmatrix} 2.5\times10^9 & 5.0\times10^6 & -1.6\times10^8 \\ 5.0\times10^6 & 7.4\times10^9 & -1.2\times10^9 \\ -1.6\times10^8 & -1.17\times10^9 & 1.9\times10^9 \end{bmatrix}$$
$$Eigenvlaues = [2.8\times10^9, 2.4\times10^9, 2.9\times10^7]$$

(b)

FIGURE 2. The second-order central moment matrixes and eigenvalues of different poses.

special model pose. Therefore, the study supposes there is an ideal pose of the model in space, whose second-order central moments matrix is diagonal. The pose normalization is the transformation of the model from any pose to the ideal pose.

The ideal pose of the model is defined as $M_0 = f(\mathbf{v})$, $\mathbf{v} \in \mathbb{R}^3$, and its second-order central moment matrix is

$$C_{M_0} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_3 \end{bmatrix}. \tag{9}$$

The pose after a transformation of matrix named R from the ideal pose M_0 is named M_1. The pose M_1 can be regarded as a new model different from the M_0 since their data representations are different in computers. So the M_1 could also be represented by a function $g(\mathbf{v})$, $\mathbf{v} \in \mathbb{R}^3$ like M_0. Because M_1 is transformed from M_0, there is a corresponding relationship between the points in M_1 and M_0. For example, for a point $\mathbf{v}_0$ in M_0, the corresponding point in M_1 is $R\mathbf{v}_0$, and the values of these two points are identical. Therefore, for any point $\mathbf{v}$ in M_0, there is the following correspondence in M_1,

$$g(R\mathbf{v}) = f(\mathbf{v}), \mathbf{v} \in \mathbb{R}^3. \tag{10}$$

According to the Equation (7), the centroid of M_1 is $\mathbf{m_1} = R\mathbf{m_0}$. According to the Equation (8) and (10),

$$\begin{aligned} C_1 &= \int_{\mathbf{v}\in M_1} (\mathbf{v}-\mathbf{m_1})(\mathbf{v}-\mathbf{m_1})^T g(\mathbf{v})d\mathbf{v} \\ &= \int_{\mathbf{v}\in M_0} (R\mathbf{v}-R\mathbf{m_0})(R\mathbf{v}-R\mathbf{m_0})^T g(R\mathbf{v})dR\mathbf{v} \\ &= R\int_{\mathbf{v}\in M_0} (\mathbf{v}-\mathbf{m_0})(\mathbf{v}-\mathbf{m_0})^T f(\mathbf{v})d\mathbf{v}R^T \\ &= RC_0R^T. \end{aligned} \tag{11}$$

According to the properties of the rotation matrix R, an orthogonal matrix, there has $R^T = R^{-1}$, so $C_1 = RC_0R^{-1}$, which is the eigendecomposition of C_1. Besides, the eigenvalues of C_1 and C_0 are equal.

From Equation (11), the matrix formed by the eigenvectors of the second-order central comment matrix of a model pose is the transformation matrix from the current pose to the ideal pose. Meanwhile, the transformation matrix should ensure that the positive axis has a greater boundary value than its negative axis.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section proposes a method for measuring the pose normalization methods, employs this method to evaluate various algorithms and examines the effect of voxel size on the effect of pose normalization.

A. THE POSE-NORMALIZATION ALGORITHM

According to section III-C, the pose-normalization algorithm is designed to automatically normalize the model pose, as shown in Algorithm 1.

Algorithm 1 Voxel Model Pose Normalization Based on Second-Order Central Moment Matrix

Input: Voxel model M

```
Read input voxel model M
Calculate the centroid of the model m based on Equation (7)
Calculate the second-order central moment matrix C based on Equation (8)
eigenvectors, eigenvalues ← eigendecompose(C)
order ← Argsort(eigenvalues)
R ← Combine_to_Matrix(eigenvector, order)
M ← Translation_Transform(M, m)
M ← Rotation_Transform(M, R)
X_max, X_min, Y_max, Y_min, Z_max, Z_min ← Bounding_Cuboid(M)
if abs(X_max) < abs(X_min) then
    M ← Reflection_Transform(M, YZ)
end if
if abs(Y_max) < abs(Y_min) then
    M ← Reflection_Transform(M, XZ)
end if
if abs(Z_max) < abs(Z_min) then
    M ← Reflection_Transform(M, XY)
end if
```

Output: Normalized pose M

B. THE EVALUATION METHOD OF POSE NORMALIZATION

The different models resulting from diverse transformations of one model are called "poses" of a model. The objective of pose normalization is to transform any pose of a model into a certain pose. Various pose-normalization approaches with various aims often result in distinct ideal poses. Therefore, it is difficult to demonstrate the practical significance of comparing the ideal poses of various approaches. However, the ideal pose derived from several poses using the same approach should be precisely the same. Therefore, the performance of the algorithm is evaluated based on the distance between normalized poses. The distance between two poses is computed using Equation (12).

$$D(M_1, M_2) = \frac{1}{N_{M_1}} \sum_{i=1}^{N_{M_1}} \min\{\sqrt{(p_i^{M_1} - p_j^{M_2})^2} | j \in [1, N_{M_2}], j \in \mathbb{N}\}, \quad (12)$$

where $p_i^{M_1}$ and $p_j^{M_2}$ are the vertices in pose M_1, M_2 respectively, N_{M_1} and N_{M_2} are the number of vertices in them, and $D(M_1, M_2)$ is the distance between pose M_1, M_2, which means the mean value of minimum Euler distances from vertices in pose M_1 to vertices in pose M_2. Furthermore, every pose is normalized to the $[-1, 1]$ space and takes the center of mass as the coordinate origin.

It is notable that a higher deviation between the poses will result in a greater distance. In other words, the mean of distances among normalized poses from various poses supports the pose-normalization method's accuracy. The study evaluates the performance using the dataset of the Engineering Shape Benchmark (ESB) [31]. This dataset is composed of CAD models classified into 3 major classes and minor classes.

C. EFFECTS OF POSE NORMALIZATION FROM DIFFERENT POSES

Through random transformations, 10 unique poses are created for each model. Then, 10 normalized poses are generated from these original poses, and distances are determined between pairings (45 distances for every model). The study compares against CPCA, NPCA and Reflective Object Symmetry (ROSy) method [20] and shows the result in Figure 3. For pre-processing 3D models, these techniques are extensively used in a variety of situations.

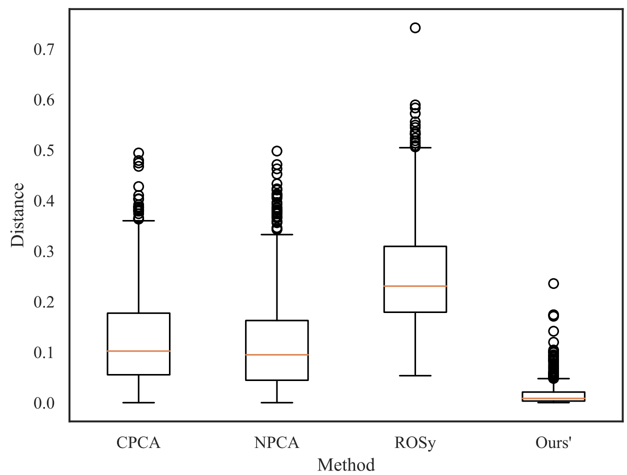
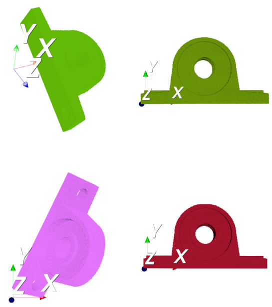


FIGURE 3. The performance of pose-normalization methods.

There is little disparity between the minimum distances of various approaches, but a substantial disparity between their maximum distances. The best result that any method can obtain is close to 0, meaning that the normalized poses are identical. However, for the majority of models, there is still a maximum variance of 0.25 between normalized poses.

The results demonstrate that the proposed method performs more consistently on a variety of CAD models than existing methods. Taking the outliers into account, the range does not surpass 0.2.

The ROSy method determines an ideal pose by continuously rotating the 3D model by a certain angle, usually 1° to 2°, to find the extreme value of the feature. Affected by the rotation accuracy, its overall error value is much higher than other methods. The PCA method is still the most popular pose-normalization method in 3D model preprocessing. It is the statistic of mesh vertices and normal vectors that make up the 3D model but lacks internal entity information of the 3D model, which explains why its overall error is higher than that of the voxel method. By analogy, supplementing surface information through volume and mass additions may endow PCA with an optimal result.



$$\begin{bmatrix} 2.6\times10^9 & -1.5\times10^6 & 4.0\times10^6 \\ -1.5\times10^6 & 2.2\times10^9 & 5.0\times10^6 \\ 4.0\times10^6 & 5.0\times10^6 & 2.5\times10^7 \end{bmatrix}$$

$$\begin{bmatrix} 2.5\times10^9 & 7.0\times10^6 & 2.6\times10^6 \\ 7.0\times10^6 & 2.2\times10^9 & 2.5\times10^6 \\ 2.6\times10^6 & 2.5\times10^6 & 2.3\times10^7 \end{bmatrix}$$

FIGURE 4. The normalized poses from various original poses with second-order central moments matrix.

Normalized samples of voxel models are depicted in Figure 4. The second-order central moment matrix of the normalized pose is not identical to Equation (9). However, the non-leading diagonal elements are all 1-2 orders of magnitude lower than the eigenvalues of the matrix. Thus, the second-order central moment matrix is comparable to Equation (9) in some sense.

D. EFFECTS OF POSE NORMALIZATION OF DIFFERENT VOXEL COUNT

The size of voxels affects the number of voxels for a model, and it may also affect the accuracy of the voxel model. In the process of voxeling the model, the voxel size is determined by dividing the shortest axis of the model into equal parts, to prevent the computational trouble caused by the excessive number of voxels, and at the same time control the expression accuracy of the model.

Additionally, the paper studies the effect of voxel size on the performance of the method. In this study, every model is voxelized and pose-normalized using various voxel sizes (presented as the number of voxels along the shortest axis). The performances of the proposed method on various voxel sizes are evaluated in the same way as section IV-C. The results are shown in Figure 5. And there are samples in Figure 6.

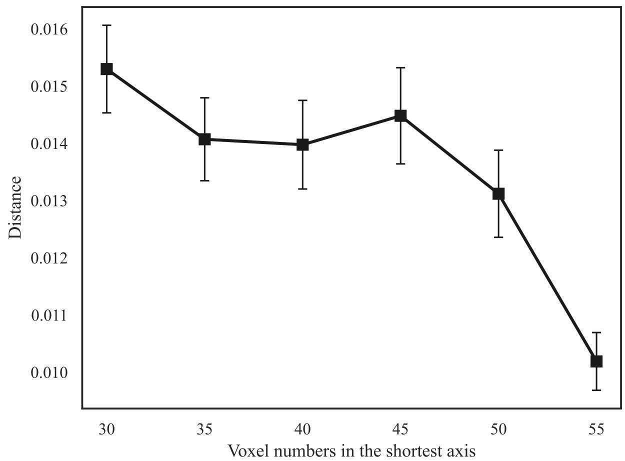


FIGURE 5. **Pose-normalization effects with various numbers of voxels in the shortest axis.**

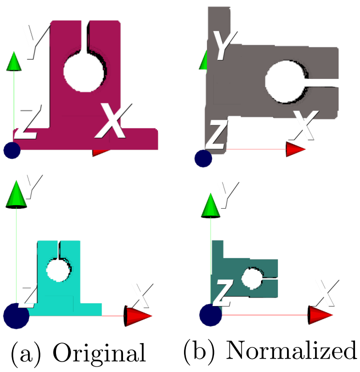


FIGURE 6. **The normalized pose with different voxel counts in the shortest axis.**

The results support that the effect of the proposed method is significantly decreasing with increasing voxel size. As the number of voxels increases, the model error distance keeps decreasing, and so does its range. Because the increase in voxel size leads to a decrease in the accuracy of the voxel model, the deformation of the model increases. Furthermore, this leads to errors in pose normalization. The voxel model is poorly refined due to its representation. For the same model, as the voxel size increases, the representation of the 3D model becomes rougher, consequently leading to more severe distortion caused during the model rotation. The result confirms that a smaller size of the voxels could give rise to a more refined representation together with an enhanced pose-normalization effect.

V. CONCLUSION

The study presents a method for pose normalization for voxel models using the second-order central moment matrix, which could transform models to their ideal poses from any poses and address the impact of the poses for feature extraction and similarity retrieval of voxel models. The voxel model takes into account the uniform distribution of model mass in space. It is also capable of expressing the structural features inside the model. However, due to the limitation of the voxel model expression capability, voxel models have poor accuracy. The results prove those: 1) The method could result in a stable pose-normalization effect for voxel models. The normalized pose meets the definition of the ideal pose of models; 2) The voxel size significantly affects the effect of pose normalization. The pose-normalization effect is inversely proportional to the voxel size. In future studies, the authors will aim to apply the algorithm to model retrieval and casting process design.

REFERENCES

[1] *Free 3D Models, CAD Files and 2D Drawings—TraceParts*. Accessed: Mar. 30, 2023. [Online]. Available: https://www.traceparts.com/en

[2] K. Hirata and T. Kato, "Query by visual example," in *Proc. Int. Conf. Extending Database Technol.* Vienna, Austria: Springer, Mar. 1992, pp. 56–71. [Online]. Available: https://openproceedings.org/html/pages/1992_edbt.html

[3] Y.-J. Liu, Y.-F. Zheng, L. Lv, Y.-M. Xuan, and X.-L. Fu, "3D model retrieval based on color + geometry signatures," *Vis. Comput.*, vol. 28, no. 1, pp. 75–86, Jan. 2012, doi: 10.1007/s00371-011-0605-8.

[4] W. Li, G. Mac, N. G. Tsoutsos, N. Gupta, and R. Karri, "Computer aided design (CAD) model search and retrieval using frequency domain file conversion," *Additive Manuf.*, vol. 36, Dec. 2020, Art. no. 101554. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221486042030926X

[5] S. Bickel, C. Sauer, B. Schleich, and S. Wartzack, "Comparing CAD part models for geometrical similarity: A concept using machine learning algorithms," *Proc. CIRP*, vol. 96, pp. 133–138, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827121000895

[6] A. Neb, I. Briki, and R. Schoenhof, "Development of a neural network to recognize standards and features from 3D CAD models," *Proc. CIRP*, vol. 93, pp. 1429–1434, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827120305552

[7] A. S. Gezawa, Y. Zhang, Q. Wang, and L. Yunqi, "A review on deep learning approaches for 3D data representations in retrieval and classifications," *IEEE Access*, vol. 8, pp. 57566–57593, 2020.

[8] W. Nie, Y. Zhao, A.-A. Liu, Z. Gao, and Y. Su, "Multi-graph convolutional network for unsupervised 3D shape retrieval," in *Proc. 28th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 3395–3403, doi: 10.1145/3394171.3413987.

[9] J. Ma and Z. Ma, "Model retrieval with 3D radon moments," in *Proc. 12th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2019, pp. 1–5.

[10] C. Vilar, S. Krug, and M. O'Nils, "Realworld 3D object recognition using a 3D extension of the HOG descriptor and a depth camera," *Sensors*, vol. 21, no. 3, p. 910, Jan. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/910

[11] H. Kanaan and A. Behrad, "Three-dimensional shape recognition and classification using local features of model views and sparse representation of shape descriptors," *J. Inf. Process. Syst.*, vol. 16, no. 2, pp. 343–359, 2020. [Online]. Available: https://koreascience.kr/article/JAKO202013965594347.page

[12] D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen–Loeve transform and spherical harmonics," in *Proc. IEEE 4th Workshop Multimedia Signal Process.*, Oct. 2001, pp. 293–298.

[13] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai, "Description of shape information for 2-D and 3-D objects," *Signal Process., Image Commun.*, vol. 16, nos. 1–2, pp. 103–122, Sep. 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596500000205

[14] D. V. Vranic, "Desire: A composite 3D-shape descriptor," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, p. 4.

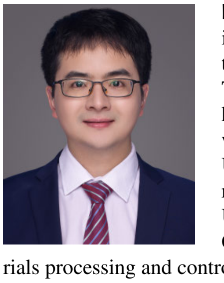
[15] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognit.*, vol. 40, no. 9, pp. 2437–2452, Sep. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320307000106

[16] M. Chaouch and A. Verroust-Blondet, "A novel method for alignment of 3D models," in *Proc. IEEE Int. Conf. Shape Modeling Appl.*, Jun. 2008, pp. 187–195.
[17] H. Guo, X. Ma, Q. Ma, K. Wang, W. Su, and D. Zhu, "LSSA_CAU: An interactive 3D point clouds analysis software for body measurement of livestock with similar forms of cows or pigs," *Comput. Electron. Agricult.*, vol. 138, pp. 60–68, Jun. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169917301746
[18] R. Yu, X. Wei, F. Tombari, and J. Sun, "Deep positional and relational feature learning for rotation-invariant point cloud analysis," in *Computer Vision—ECCV 2020* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2020, pp. 217–233.
[19] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3D models," *Algorithmica*, vol. 38, no. 1, pp. 201–225, Jan. 2004.
[20] K. Sfikas, T. Theoharis, and I. Pratikakis, "ROSy+: 3D object pose normalization based on PCA and reflective object symmetry with application in 3D object retrieval," *Int. J. Comput. Vis.*, vol. 91, no. 3, pp. 262–279, Feb. 2011, doi: 10.1007/s11263-010-0395-x.
[21] K. Sfikas, T. Theoharis, and I. Pratikakis, "Pose normalization of 3D models via reflective symmetry on panoramic views," *Vis. Comput.*, vol. 30, no. 11, pp. 1261–1274, Nov. 2014.
[22] H. Yuan, M. Y. Pang, and Z. P. Lu, "Pose normalization based on rotation transformation," *Key Eng. Mater.*, vol. 464, pp. 453–456, Jan. 2011. [Online]. Available: https://www.scientific.net/KEM.464.453
[23] H. Fu, D. Cohen-Or, G. Dror, and A. Sheffer, "Upright orientation of man-made objects," in *Proc. ACM SIGGRAPH Papers*, Aug. 2008, pp. 1–7.
[24] T. Czerniawski, M. Nahangi, S. Walbridge, and C. Haas, "Automated removal of planar clutter from 3D point clouds for improving industrial object recognition," in *Proc. Int. Symp. Autom. Robot. Construct. (IAARC)*, vol. 33. Auburn, AL, USA: IAARC Publications, Jul. 2016, pp. 357–365. [Online]. Available: https://www.iaarc.org/publications/2016_proceedings_of_the_33rd_isarc_auburn_usa/automated_removal_of_planar_clutter_from_3d_point_clouds_for_improving_industrial_object_recognition.html, doi: 10.22260/ISARC2016/0044.
[25] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," Oct. 2017, *arXiv:1604.03351*.
[26] L. Vancoillie, K. Houbrechts, J. Vignero, M. Keupers, L. Cockmartin, N. W. Marshall, and H. Bosmans, "The creation of a breast cancer voxel model database for virtual clinical trials in digital breast tomosynthesis," *Proc. SPIE*, vol. 11595, pp. 173–180, Feb. 2021. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11595/115950P/The-creation-of-a-breast-cancer-voxel-model-database-for/10.1117/12.2581741.full
[27] A. V. Tolok, N. B. Tolok, and E. R. Batuev, "Voxel modeling of the control of prototype manufacturing with additive technologies," *Autom. Remote Control*, vol. 82, no. 3, pp. 506–515, Mar. 2021.
[28] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2088–2096.
[29] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.
[30] F. A. Sadjadi and E. L. Hall, "Three-dimensional moment invariants," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 2, pp. 127–136, Mar. 1980.
[31] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani, "Developing an engineering shape benchmark for CAD models," *Comput.-Aided Design*, vol. 38, no. 9, pp. 939–953, Sep. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001044850600100X

SHUREN GUO received the B.S. degree in materials forming and control engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently pursuing the Ph.D. degree in materials processing engineering with the School of Materials Science and Engineering. His research interests include intelligent casting, 3D sand mold printing, and sand casting process design.

XUANPU DONG received the Ph.D. degree in materials processing from the Huazhong University of Science and Technology, in 2002, with a focus on liquid metal forming. Since 2008, he has been a Professor and a Ph.D. Supervisor with the Huazhong University of Science and Technology. His research interests include liquid metal forming and intelligent casting processes.

DONG XIANG received the B.S. degree in materials forming and control engineering and the M.S. degree in materials processing engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2009, respectively, where he is currently pursuing the Ph.D. degree in materials processing engineering with the School of Materials Science and Engineering. His research interest includes computer-aided casting process designs.



HUATANG CAO received the Ph.D. degree in materials science and engineering from the University of Groningen, Groningen, The Netherlands, in 2019. Then, he continued his postdoctoral research in materials engineering with The University of Manchester, Manchester, U.K. He has been an Associate Professor in materials processing engineering with the Huazhong University of Science and Technology, Wuhan, China. His research interests include metal materials processing and controlling, casting, surface engineering, and advanced coatings.