TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 12/16 pp754–768
DOI: 10.26599/TST.2022.9010026
Volume 28, Number 4, August 2023

# Three Dimensional Metal-Surface Processing Parameter Generation Through Machine Learning-Based Nonlinear Mapping

Min Zhu, Yanjun Dong, Bingqing Shen, Haiyan Yu, Lihong Jiang, and Hongming Cai*

**Abstract:** The accuracy and efficiency of three-dimensional (3D) surface forming, which directly affects the cycle and quality of production, is important in manufacturing. In practice, given the uncertainty of metal plate springback, an error exists between the actual plate and the target surface, which creates a nonlinear mapping from computer aided design models to bending surfaces. Technicians need to reconfigure parameters and process a surface multiple times to delicately control springback, which greatly wastes human and material resources. This study aims to address the springback control problem to improve the efficiency and accuracy of sheet metal forming. A basic computation approach is proposed based on the DeepFit model to calculate the springback value in 3D surface bending. To address the sample data shortage problem, we put forward an advanced approach by combining a deep learning model with case-based reasoning (CBR). Next, a multi-model fused bending parameter generation framework is devised to implement the advanced springback computation approach through surface data preprocessing, CBR-based model matching, convolution neural network-based machining surface generation, and bending parameter generation with a series of model transformations. Moreover, the proposed approaches and the framework are verified by considering saddle surface processing as an example. Overall, this study provides a new idea to improve the accuracy and efficiency of surface processing.

**Key words:** 3D surface; point-cloud; machine learning; case-based reasoning; industrial software

## 1 Introduction

The three-dimensional (3D) plate bending process is an important procedure in high-end equipment manufacturing; it modifies the shape of metal plates with steel or alloy materials. The bending accuracy and efficiency directly affect the quality of ships. A key step is a generation of processing parameters through the surface development algorithm in computer aided design (CAD) models. Then, the 3D computer numerical control (CNC) plate bending machine can form the ship plate into a specific geometry by multi-point cold pressing. However, given the uncertainty of springback, an error exists between the target and actual shapes of the ship plate[1]. Thus, in practice, technicians have to reduce the gap with a trial-and-error approach. They modify configuration parameters and bend the ship plate multiple times. However, this approach has low efficiency, costs a huge amount of manpower and material resources, and still lacks a guaranteed bending accuracy.

For bending accuracy improvement, the most effective method is the modification of configuration parameters by considering the springback of the plate[2]. The bending result can meet the engineering requirement in geometry shape and forming accuracy. Some forming parameter optimization approaches have been

• Min Zhu, Yanjun Dong, Bingqing Shen, Lihong Jiang, and Hongming Cai are with the School of Software, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: ericzhumin@sjtu.edu.cn; 15723633724@sjtu.edu.cn; sunniel@sjtu.edu.cn; jianglh@sjtu.edu.cn; hmcai@sjtu.edu.cn.
• Haiyan Yu is with the School of Mechanical Engineering, Donghua University, Shanghai 200240, China. E-mail: yuhy@dhu.edu.cn.
∗ To whom correspondence should be addressed.

Manuscript received: 2022-03-22; revised: 2022-06-27; accepted: 2022-07-01

proposed in existing studies. For example, Feng et al.[3] adopted the sequential approximation multi-objective optimization method to obtain the optimal variable blank-holder force in sheet metal forming under the condition of interval uncertainty. Gao et al.[4] classified surface morphing into algebraic and free-form morphing and obtained control points by data mining. Likewise, in the case of plate bending, we need to calculate the optimal configuration parameters of processing points from a CAD model to obtain qualified bending results.

However, the mapping relation from a CAD model to bending parameters is nonlinear and difficult to predict because many factors can cause springback and surface deformation, e.g., material, thickness, geometry, and a variety of other factors. With these factors, surface mapping becomes nonlinear, and thus, the mapping relation with traditional physical and geometric models is hard to describe. Introducing machine learning into computational methods is an important solution to the causal and complex nonlinear mapping problem. However, two additional challenges arise in the generation of 3D plate bending parameters.

The first challenge is the training data for springback computation. In general, high-end equipment manufacturing (e.g., shipbuilding) has the difficulty of using a large number of sample types but a small size of the data sample type, which affects the accuracy of machine learning results. Therefore, new methods are necessary to maximize the use of existing samples while ensuring the accuracy of machine learning results.

The second challenge is the models for bending parameter generation. To approach a machine learning-based solution, scientists need to convert CAD models to machine learning models. However, CAD models are normally composed of a large number of vertices, edges, and faces and are thus hard to modify due to the high storage space cost, low computational efficiency, and low consistency after deformation. Moreover, CAD models in practice have different sizes, positions, attitudes, and point distributions on the surface, increasing the difficulty of machine learning.

To meet the above challenges, this study uses a point cloud to represent the CAD model and proposes an intelligent deep learning method of 3D plate bending parameter generation based on the point-cloud model. As a result, the model features can be extracted with case-similarity calculation, which can improve the accuracy of model-parameter mapping. Moreover, the feasibility and practicability of the proposed method are verified by studying a case of saddle surface bending in shipbuilding. Overall, this paper has the following contributions.

(1) It proposes a deep learning-based intelligent approach for 3D plate bending parameter prediction by leveraging the spatial features of point-cloud to establish a nonlinear mapping from a CAD model to bending parameters.

(2) It proposes a case-based reasoning (CBR) based model matching algorithm to improve the mapping accuracy with multi-type small-sized samples.

(3) It devises a multi-model fused bending parameter generation (MMFBPG) framework to apply the proposed approach to real production with data preprocessing and parameter generation methods.

(4) It provides a case study on ship plate bending to show the feasibility and efficiency of the proposed approach and the devised framework.

The paper is organized as follows. Section 2 introduces the related work. Section 3 briefly introduces the 3D CNC plate bending process and proposes a 3D processing parameter generation framework. Section 4 devises the MMFBPG approach. Section 5 provides a case study and Section 6 compares the proposed approach with existing ones. Finally, Section 7 concludes the study.

## 2 Related Work

The related research for point-cloud-based curved plate transformation can be divided into model construction and enhancement, model matching, and transformation.

### 2.1 Model construction and enhancement

Model construction and enhancement based on point-cloud aim at standardizing the point-cloud model with the predefined rules for subsequent processing. In general, it is applied to point-cloud surface reconstruction. At present, three main classes of point-cloud surface reconstruction approaches, including mesh-based point-cloud reconstruction, implicit surface reconstruction, and parametric surface reconstruction, are available.

Mesh-based point-cloud reconstruction uses mesh approximation instead of surfaces. Hou and Gu[5] studied the triangular mesh generation of scattered points in space and realized uniform mesh reconstruction using a micro incremental method. Wang et al.[6] used the angle condition of adjacent triangles and the disjoint condition of triangles to generate surfaces from a triangular mesh. The mesh model has strong flexibility and good adaptability to the surface boundary, but it also has many shortcomings, including a large number of mesh patches, large memory resources, and low reconstruction accuracy.

Implicit surface reconstruction relies on implicit functions. In this class, polynomial and radial functions have been used to describe the point-cloud surface. Qin et al.[7] weighted the curvature of the surface reconstructed by the Poisson function and smoothed the surface by setting the curvature threshold. Implicit surface reconstruction can well preserve surface details, but it is easily disturbed by the noise of point-cloud data. In addition, the reconstructed surface edge is easily deformed.

In parametric surface reconstruction, B-spline surface[8] and non-uniform rational B-spline (NURBS) surface[7] are the most representative methods for parametric surface fitting. A B-spline surface has good smoothness and stable geometric topology information for fitting object surfaces, but it needs to be based on a rectangular mesh topology. NURBS introduces a weight factor to improve the precision of surface expression and analysis. NURBS surface fitting usually adopts a rectangular parameter domain, which has an evident quadrilateral boundary in the target model, when the contour lines in point-cloud data are orderly and regularly distributed. The NURBS technology has been successfully applied to plate surface design for the generation of feature lines and plate surfaces. Thus, NURBS is suitable for our surface reconstruction.

### 2.2 Model matching and transforming

Point-cloud is a geometric data structure with four characteristics. First, the underlying shape is not affected by the arrangement order of points in the point cloud (irregularity). Second, a point-cloud model contains spatial information in which the characteristics of each point are closely related to its adjacent points (domain correlation). Third, the object represented by point-cloud data after spatial transformation (e.g., rotation and translation) is invariable (invariance). Last, the distribution of points is often uneven when the point cloud describes the actual object (uneven distribution). Given the above characteristics, the point-cloud model matching approaches, including convolution neural network (CNN) based and PointNet-based approaches, show different performances.

Deep neural networks exhibit good performance in image recognition and classification[9]. Zhou et al.[10] adopted CNNs for feature extraction and classification of 3D images. However, different from images, CNNs cannot obtain the spatial position information of the point cloud due to their irregularity and invariance characteristics, leading to inaccurate convolution results in the point-cloud model. To solve this problem, scientists use most of the existing methods to convert the point-cloud model into regular image sets[11] or voxelization[12]. Then, the deep learning models for image processing can be applied to point-cloud data. However, these methods cannot fully learn the spatial information of the point cloud. Meanwhile, owing to a large number of points, the amount of calculation for point-cloud models is considerably larger than that in image processing.

On the other hand, Qi et al.[13] proposed a deep learning framework for point-cloud, called PointNet, which can be directly applied to point-cloud data and has achieved good performance in model classification and segmentation. To better learn local features, Qi et al.[14] proposed the PointNet++ framework. PointNet++ can extract local features of point-cloud data through point sampling and clustering. Then, it continuously learns the local features of point-cloud data and, finally, the deep features. PointNet and PointNet++ have been applied to 3D object detection[15] and semantic segmentation[16] with good results. Wang et al.[17] proposed the dynamic graph CNN (DGCNN) model, which has a similar network structure to PointNet. DGCNN uses Edge-Conv to obtain global and local features of point-cloud data, which distinguishes it from PointNet.

In short, although AI methods are commonly used for image and video disposal, the intelligent method related to point-cloud disposal still needs considerable work for manufacturing applications.

## 3 Background and Framework

### 3.1 Overview of 3D CNC plate bending

Multi-point forming technology is widely used in bending plate processing. The 3D CNC plate bending machine has upper and lower dies, and both contain multiple discretely arranged indentors[18]. By adjusting the height of indentors in the upper and lower dies, the 3D CNC plate bending machine forms an envelope surface corresponding to the target curved surface shape and bends the plate through a multi-point press forming method. In practice, after one bending, the height of the indentors in the upper and lower die must be adjusted based on the measured error of the bent plate to form a new envelope surface and perform another bending to approach the target shape.

Springback is inevitable in the process of plate bending. When the external pressure from the indentors of a plate bending machine disappears, elastic deformation will recover, resulting in a deformation opposite to the bending direction. The extent of deformation is determined by the amount of springback. Springback is closely related to many factors, such as shape forming and material characteristics. Thus, the influence of springback on plate bending is difficult to predict.

### 3.2 Framework

To implement the intelligent computation of springback control, we propose an MMFBPG framework to address the nonlinear mapping problem of bending processing parameter generation in 3D plate bending. As shown in Fig. 1, the overall framework contains four steps, including data preprocessing, CBR-based model matching, machine learning-based machining surface generation, and bending machine processing parameter generation.

In the MMFBPG framework, the process of CBR-based model matching and machine learning-based machining surface generation collectively implements the springback computation approach (multi-model fusion). In addition to the multi-model fusion process, we add rule-based data preprocessing and bending machine processing parameter generation to meet the needs in practice. In rule-based data preprocessing, the CAD model of the plate to bend is exported to the point-cloud model and processed to obtain the uniform spatial constraint to facilitate the subsequent model matching and machine learning. In bending machine parameter generation, the machining surface of a plate is configured, and the manufactured surface is generated based on the position of the upper and lower dies in a 3D CNC bending machine. With the MMFBPG framework, the generated processing parameters can be directly applied to the 3D CNC bending machine.

## 4 MMFBPG Approach

Adjusting the shape of dies is the most effective way to control processing parameters, and it can be achieved by adding an appropriate amount in the opposite direction of springback. To resolve this issue, in this section, we introduce a springback calculation method based on deep neural networks and improve its accuracy with CBR.

### 4.1 Data preprocessing for 3D surfaces

In practice, the models of different designs may differ in size, spatial state (position and attitude), and point distribution on the surface, but they may describe similar shapes. Thus, to ensure machine learning accuracy, scientists need data preprocessing to generate a uniform model for CBR in terms of coordinates and point distribution. We proposed a rule-based model enhancement approach, which includes coordinate transformation, surface reconstruction, and point collection, for 3D surface data preprocessing to establish a uniform spatial constraint.

After exporting a CAD model to a point-cloud model, the coordinate transformation function translates, rotates, and zooms the point-cloud model to generate a uniform

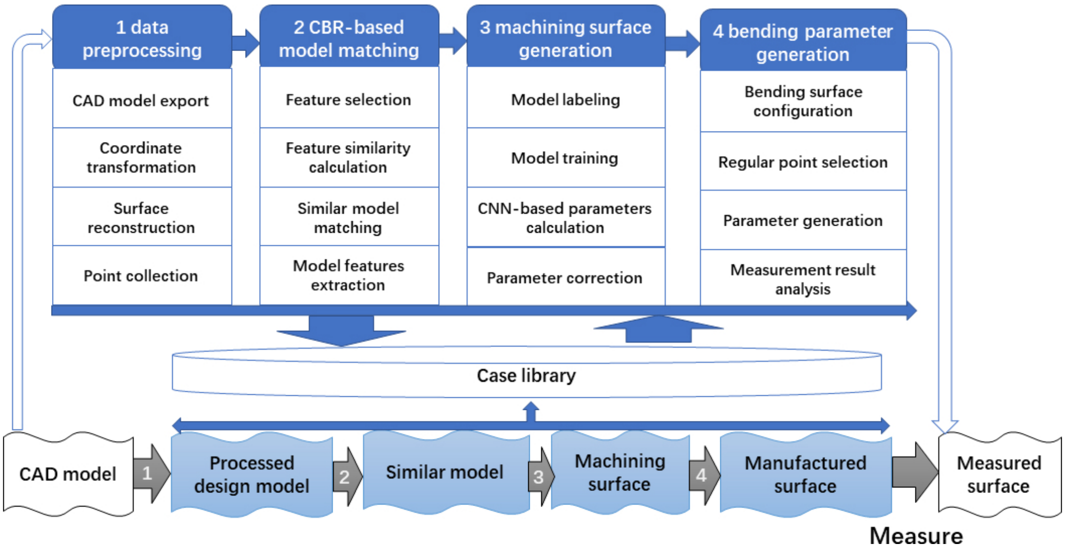


**Fig. 1 Diagram of multi-model fused bending parameter generation framework.**

point-cloud model as a processed design model (PDM). Specifically, first, the oriented bounding box (OBB) of the point-cloud model is calculated to describe the spatial position and attitude of the model, and based on the eight vertices on the bounding box, we can redefine the surface coordinates shown in Fig. 2.

Specifically, based on the surface area of the bounding box, we select the face with the largest area and close to the $xy$ plane as the $x'y'$ plane to ensure that the $z$ values of all points on the point cloud are non-negative. We connect the midpoints on both sides as the $x'$ axis. Accordingly, we re-select the $z'$ and $y'$ axes. After the coordinate transformation, the point cloud is enlarged or reduced in proportion. As a result, the surface projection on the $X'Y'$ plane is within a specified range, in which the longitudinal projection length is between 1600 mm and 3200 mm.

Next, the point-cloud model is reconstructed from a series of points as a smooth surface. We adopt the NURBS surface reconstruction method[19], in which points are selected interactively to reconstruct NURBS curves.

As the input of the neural network for springback computation requires the same scale of point-cloud models, the points must be recollected to ensure the same number of points on each point cloud. Meanwhile, the points are retrieved uniformly from the surface to avoid ignoring the information on the surface during point collection. Thus, in the last step, we project the surface described by the point cloud onto the $xy$ plane, collect 100 equal parts in the $x$- and $y$-axis directions, and collect the corresponding points on the surface to form a 100-point × 100-point point-cloud model.

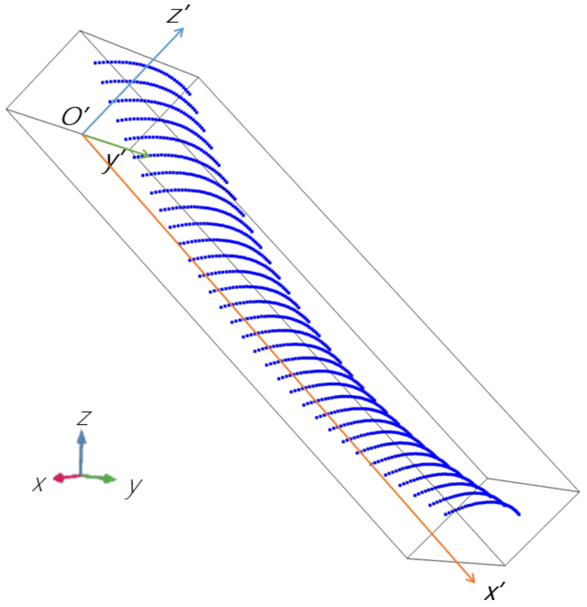


**Fig. 2 Schematic diagram of the coordinate transformation from *xyz* to *x′y′z′*.**

### 4.2 CBR-based model matching

Although deep neural networks can be used to describe nonlinear mapping, their accuracy is affected by the training data. To improve the reliability and interpretability of deep learning results, we proposed a CBR-based model matching method to obtain similar cases.

We used the PointNet++ and OBB-based feature extraction models to extract the feature of the point-cloud model (feature selection), calculated the feature similarity of the two models (feature similarity calculation), matched models based on similarity (similar model matching), and obtained all informations related to springback calculation of similar models (model feature exaction).

A large number of features on the shape of a surface prevent the traditional feature selection from extracting complete shape information. Therefore, in feature extraction, we used PointNet++ and OBB-based feature extraction models to extract the feature vector of the point-cloud model. Notably, feature extraction is performed on the point-cloud model.

Based on the OBB model, we obtain the ratio of length, width, and height as the characteristic attribute of the point-cloud model and give height the value of 1 to calculate the ratio of length and width to height.

Based on the PointNet++ model, we can extract the local and global features of the point-cloud model, which contains the shape information of the surface. $Z$-score standardization (Eq. (1)) was used to preprocess the coordinates of points in the point-cloud model.

$$x^* = \frac{x-\mu}{\sigma} \tag{1}$$

where $x$ and $x^*$ represent the coordinate values before and after standardization, respectively, $\mu$ is the mean of the original coordinates, and $\sigma$ is the standard deviation of the original coordinates.

The PointNet++ model samples and clusters the point-cloud model based on the coordinate position and carries out hierarchical feature extraction. In the PointNet++ model, each feature layer will extract the corresponding attributes based on the clustering of point coordinates.

The last layer extracts the feature attributes from the feature layer and inputs them into the full connection layer for classification. We used the penultimate layer of the last fully connected layer to calculate the similarity of bending cases. In our PointNet++ model, the dimension of the penultimate layer is 256.

Thus far, we have obtained two feature vectors to measure similarity. Next, based on the feature vectors, similarity-based model matching can be performed. For the length-to-width-to-height ratio, we can directly obtain the similarity between models by the Euclidean distance formula $d(u_1, u_2)$, $\mathrm{sim}(u_1, u_2) = 1/(1+d(u_1, u_2))$, with any eigenvalue of $u_1$ and $u_2$. For the multi-dimensional feature vector, the cosine distance of the feature vector in Eq. (2) is used to obtain the feature similarity through weighted average:

$$\mathrm{sim}(u_1, u_2) = \sum_{k=1}^{m} \mathrm{sim}_{\mathrm{num}}^{(u_1\times m_k, u_2\times m_k)}/n + \mathrm{sim}_{\cos}(u_1\times w, u_2\times w) \quad (2)$$

We then calculated the average of the two feature similarities to obtain the final feature similarity $C$. As the surface type is a known quantity in practice, when searching for similar cases, we only searched for them within the same category of surfaces to improve the accuracy. Specifically, we searched the $k$ nearest neighbors to retrieve the $k$ most similar cases. After finding similar cases, the corresponding actual springback $S$ can be obtained.

## 4.3 Machining surface generation

### 4.3.1 CNN-based parameter calculation

To address the nonlinear mapping problem, this work proposes a deep learning method for calculating the springback in 3D plate bending by fitting the nonlinear mapping relationship between the point-cloud model of various surfaces and their springback values.

We adopted the improved model of DeepFit[20] to achieve nonlinear mapping. DeepFit uses PointNet to extract features and a multi-layer perceptron model to predict the weight of points in the fitting. The springback at a point in the point cloud is closely related to the shape of the surface around the point, and it can be determined by the coordinates of neighbors. Thus, based on the weight extraction of DeepFit, we adopted a more efficient way by changing the learning goal to learn the springback of every point. Figure 3 shows the design of the improved DeepFit model. Through the improved DeepFit model, we can input the point-cloud model to obtain the springback of each point. In the improved DeepFit model, the $K$ nearest neighbor points ($S_i$) of the point-cloud model are inputted to the PointNet network, which outputs the global point-cloud representation $\mathrm{tt}(S_i)$. In addition, the local representation of each point $P_j \in S_i$ is extracted from the middle layer to yield $g(P_j)$.

These representations are fed into the multi-layer perceptron. The output of this network is the weight of each point. After obtaining the weight matrix $\boldsymbol{W}$, the springback compensation ($\mathrm{d}z$) is calculated by $\mathrm{d}z = W\cdot Z$, in which $Z$ is the set of coordinate $z$ values in the $k$ neighbors. The loss function is the mean square loss function between $\mathrm{d}z$ and the springback of the center point ($\mathrm{d}z^t$), as shown in Eq. (3).

$$\mathrm{loss}(\mathrm{d}z, \mathrm{d}z^t) = (\mathrm{d}z - \mathrm{d}z^t)^2 \quad (3)$$

### 4.3.2 Parameter correction

In Section 4.2, we propose a CBR-based model matching method to obtain similar models and their corresponding springback $S$. Based on these models, we modify the fitting results $Z$ of the target case and evaluate the confidence in this section.

We predicted the springback of similar cases ($\boldsymbol{S}' = (S'_1, S'_2, \ldots, S'_k)$) and the target case ($Z^{tc}$) separately. With the known surface types in practice, we trained an improved DeepFit model for each type to improve the

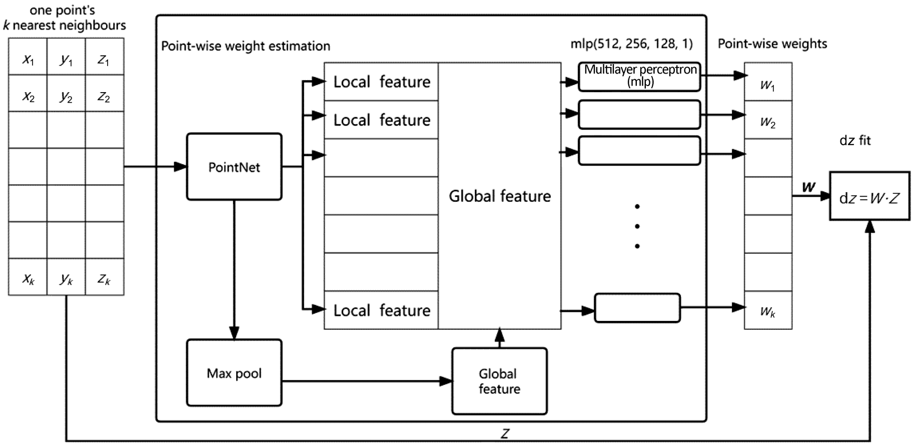


**Fig. 3 Diagram of improved DeepFit model.**

accuracy. Then, we inputed the point-cloud models of the target case and similar cases into the improved model to obtain the springback prediction results.

The predicted springback of the target case $Z^{tc}$ can be revised on the basis of similar cases. Algorithm 1 describes the process of springback correction.

In the algorithm, we first compared the prediction result $S'$ with the actual springback of the point-cloud model in similar cases (recorded as $\boldsymbol{S} = (S_1, S_2, \ldots, S_k)$) to calculate the springback error of each point (Line 2).

After obtaining the error of springback of similar cases, we calculated the adjustment value $E$ of the springback using the similarity between similar cases and the target case (Eq. (4)):

$$E = \sum_{i=1}^{k} \text{sim}\left(C_i, \text{TargetCase}\right) \times e_i / k \tag{4}$$

**Algorithm 1 Parameter correction**

```
Input: Prediction springback of target case $Z^{tc}$, prediction springback of similar cases $S'$, actual springback of similar cases $S$, and similarity of target case and similar cases $C$;
Output: Corrected springback of target case $Z^t$;
1:  for $i$ in $S$ do
2:      $e_i \leftarrow (S_i - S'_i)^2$;
3:  end for
4:  $E \leftarrow$ adjustment value of springback based on $e_i$ and $C$;
5:  CD $\leftarrow$ the confidence base on $C$;
6:  if CD $> 0.8$ then
7:      $Z^t \leftarrow Z^{tc} + E$;
8:  else
9:      $Z^t \leftarrow Z^{tc}$;
10: end if
11: return $Z^t$
```

where $C$ is a similar case. Meanwhile, we calculated the confidence CD based on the similarity using Eq. (5), which can be used to measure the reliability of the deep learning model in springback prediction.

$$\text{CD} = \sum_{i=1}^{k} \text{sim}(C_i, \text{TargetCase}) / k \tag{5}$$

Particularly, if the CD value is above 0.8, similar cases are highly accurate and have a high reference value. Therefore, the adjustment value $E$ is added to $Z^{tc}$ to obtain the final springback $Z^t$. If the reliability is below 0.8, then the prediction result of the case is insufficient and cannot be applied to the prediction result of the target case. Therefore, when the accuracy of the neural network is high, $Z^{tc}$ is directly considered as the final springback $Z^t$. With the increase in data and equipment types, the management of heterogeneous and various data has become important[21] . Therefore, we needed to build a unified case library before CBR-based model matching and CNN-based surface generation.

The most important knowledge of a historical bending case includes the type, thickness, and material of a plate because they are closely related to the springback of bending. Meanwhile, the feature vectors of historical plate data are calculated in advance for fast retrieval. The above knowledge is converted to data and stored in the case library. Figure 4 shows the case library structure.

Currently, a bending case includes the original design model (ODM), PDM, machining surface, manufactured surface, and measured surface. The ODM includes the point-cloud model exported from the CAD model, and it describes the target surface shape of a curved plate

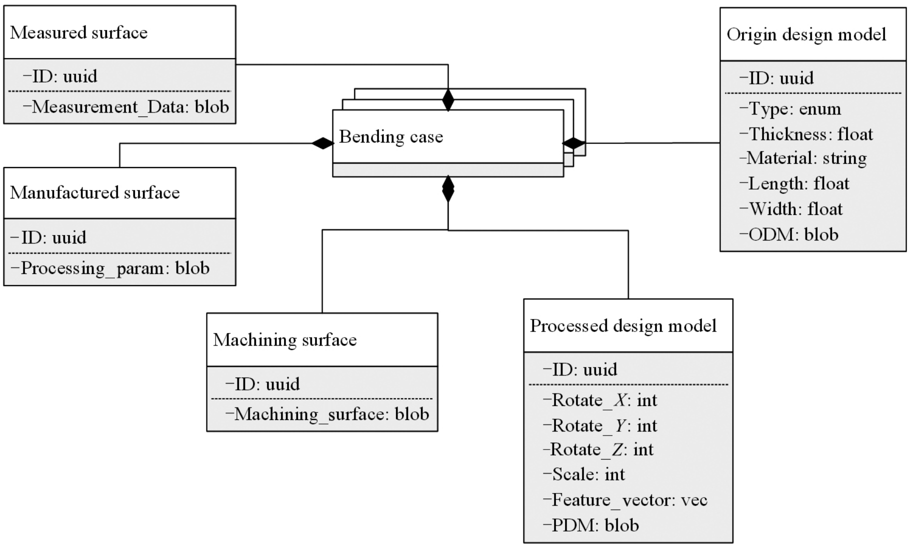


**Fig. 4 Diagram of the structure of the case library.**

and the basic parameters. The PDM describes the data after a spatial coordinate transformation, feature vector, and parameters of rotation and scaling. The machining surface contains the surface data with springback correction. The manufactured surface comprises the actual processing parameters used by the 3D CNC bending machine. The measured surface consists of the actual surface after bending.

Moreover, a DeepFit model is trained for each type of plate, and a PointNet++ model is trained for all plate types. The DeepFit models and a PointNet++ model are stored separately outside the case library. We can use a different DeepFit model based on the plate type.

### 4.4 Bending parameter generation

After calculating the springback value of a 3D model, bending parameters can be generated for 3D CNC bending machine control.

As the 3D surface data have been converted to a uniform point-cloud model in rule-based 3D surface data preprocessing, the machining surface must be restored to its original scale by enlargement and reduction in proportion based on the scale stored in the case library.

Based on the position of the upper and lower dies, four types of bending parameters need to be generated. These parameters include the row and column where the indentor is located, the height of the indentor in the lower die, and the deviation between the heights of the indentors in the lower and upper dies. After reconstructing the surface, we converted the calculated *XY* value of the 3D CNC plate to the processing parameter of a bending machine by collecting the points from the generated surface based on the position of indentors.

Finally, by saving the obtained processing parameters to a configuration file with a required format, it can be directly imported into the 3D CNC plate bending machine for surface forming. After machining, the system collects the shape information of the actually formed surface through measurement and saves it into the case base as the measured data for future computation.

## 5 Case Study

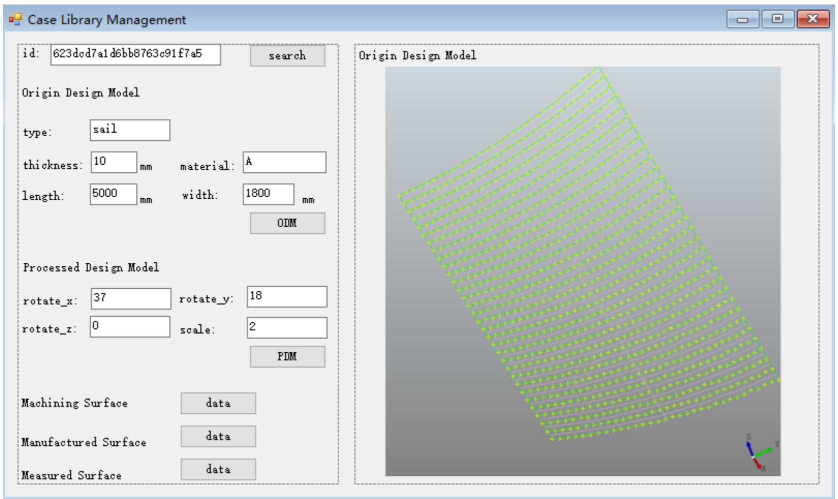
### 5.1 Study on ship plate bending

This section studies the case of the 3D bending of saddle ship plates to show the feasibility of the proposed MMFBPG framework in 3D bending parameter generation.

Based on the proposed method, we first built a case library by obtaining the data of multiple types of ship plates from the shipyard. Figure 5 shows the GUI of our developed case library, which contains the information on a bending case history.

Based on the quantity and distribution of each type, we select the six main types of ship plate data and categorize them into cylindrical, conical, sail, saddle, transverse bending and twisting, and transverse wave bending plates. The rest are classified into other categories. As each category contains 200 copies of ship plate data, we obtain 1400 copies of ship plate data overall. Each copy of ship plate data includes the corresponding CAD model and processing parameters of the ship plate. The CAD model is exported as data in point-cloud format. Figure 6 illustrates the ODM of the six types of ship plate surfaces.

Then, the ODM and processing parameters are respectively converted to the PDM and machining



id: 623dcd7a1d6bb8763c91f7a5

Origin Design Model

type: sail

thickness: 10 mm material: A

length: 5000 mm width: 1800 mm

Processed Design Model

rotate_x: 37 rotate_y: 18

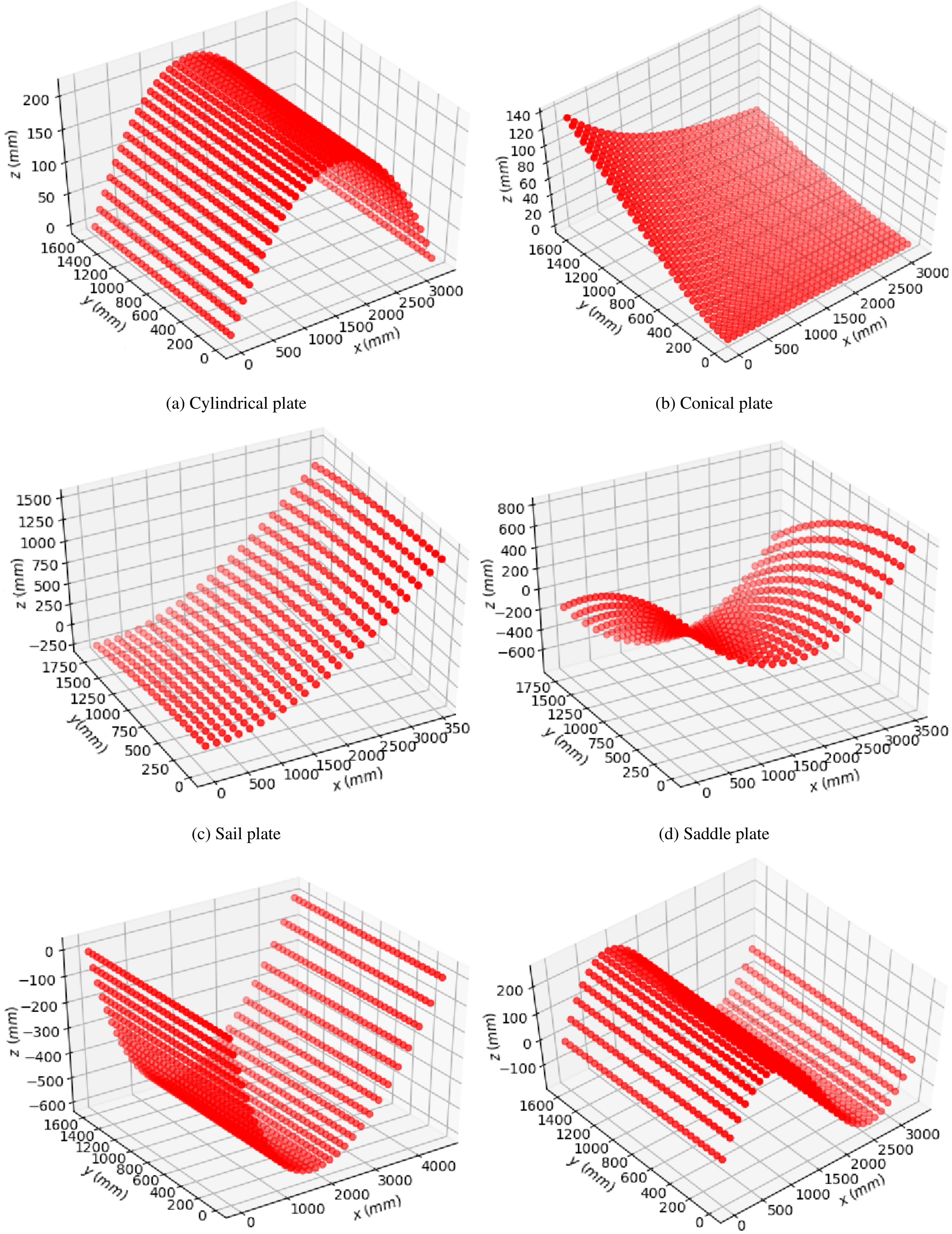
rotate_z: 0 scale: 2

Machining Surface

Manufactured Surface

Measured Surface

**Fig. 5 Software GUI of the case library.**



(a) Cylindrical plate

(b) Conical plate

(c) Sail plate

(d) Saddle plate

(e) Lateral bending and twisting

(f) Transverse wave bending

**Fig. 6 Six typical ship plate models.**

surface through data preprocessing, which are then stored in the case library for the training of the PointNet++ model and the DeepFit models.

In this paper, the effectiveness of the proposed method is verified by the 3D saddle parts and the curved parts of a hull curved outer plate, which is commonly used in production. The selected saddle parts have the following features. The longitudinal projection length is 3200 mm; the transverse projection length is 1600 mm. And the plate thickness is 25 mm.

The selected curved parts have the following features. The longitudinal radius of the curvature is 8000 mm, and the bending direction is positive. The transverse radius of the curvature is 4000 mm, and the bending direction is negative. We used the CAD models of saddle ship plates as the input data and carried out the MMFBPG process with the following steps.

**Step (1) Data preprocessing**

The CAD model was exported as the ODM in point-cloud format. After extracting the OBB of the point-cloud model, the central axis of the bounding box was extracted based on predefined rules to determine a new coordinate system. Within the new coordination system, the coordinate transformation function and the proportional scaling function were carried out to generate the OBB of the point-cloud model with unified spatial constraints (Fig. 7). The length of the long rectangular side was between 1600 nm and 3200 mm. Then, the NURBS reconstruction method was used to reconstruct the point-cloud model, and points were evenly obtained on the reconstructed surface to form a 100 × 100 lattice. As a result, we acquired the design data of the saddle ship plate (Fig. 8).

**Step (2) Model matching**

In this step, we inputed the PDM into the PointNet++ model to obtain the feature vector, calculated the length-width-height ratio, and obtained the feature value after synthesizing both feature vectors. For model matching, we calculated the similarity with the design data from the case library and obtained ten similar design data of the saddle ship plate. The maximum, minimum, and average similarities with the target cases are 0.865, 0.795, and 0.827, respectively. Figure 9 shows the cas comparison with the highest similarity.

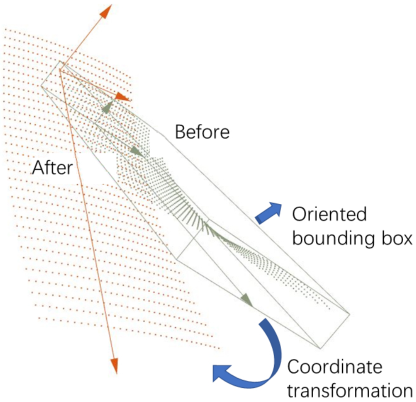


**Fig. 7 Coordinate transformation based on the central axis.**

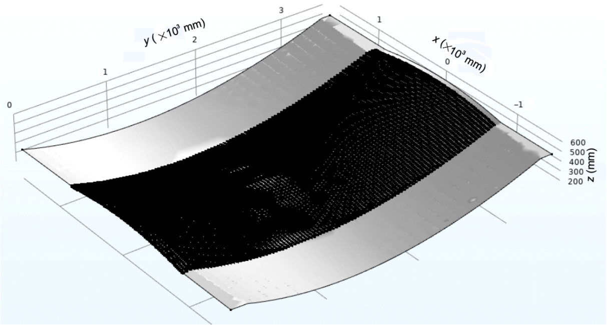


**Fig. 8 Non-uniform rational B-spline-based surface reconstruction and regular point selection.**

**Step (3) Machining surface generation**

To measure the reliability of similar cases, we first calculated their confidence based on their similarity. The confidence of similar cases is 0.827, which shows that similar cases are accurate and have a high reference value for target cases. Therefore, we inputed the PDMs of the target case and similar cases into the DeepFit model for the saddle ship plate to obtain the springback prediction results, namely, $Z^{tc}$ and $S_t$. Then, we compared the actual ($S$) and predicted springback ($S_t$) to obtain the springback adjustment value $E$ (Fig. 10). Finally, $E$ was added to $Z^{tc}$ to obtain the final springback to revise the design data of the machining surface (Fig. 11).

**Step (4) Bending parameter generation**

The size and distribution of the mold group for

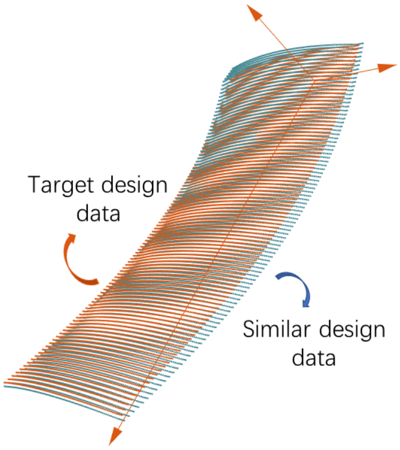


**Fig. 9 Comparison of similar cases.**

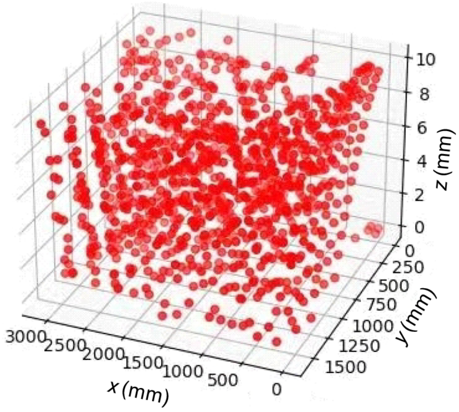


**Fig. 10 Springback adjustment values at each point of the curved surface.**

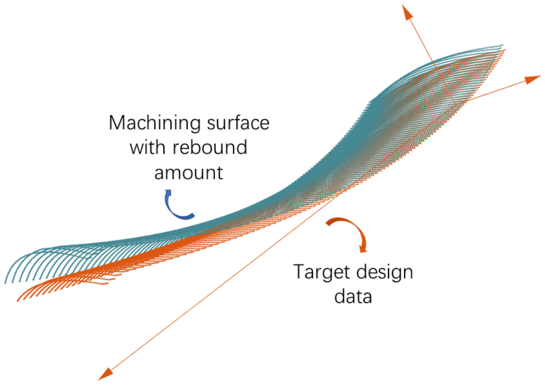


**Fig. 11 Diagram of the machined surface.**

different types of 3D CNC bending machines vary. Thus, the correct processing configuration file needs to be created from the generated machining surface for a specific bending machine.

First, the machining surface was reconstructed. According to the 20 × 20 mold group position, the coordinates of surface points were collected from the point-cloud model. Specifically, the height of the lower mold was obtained as the lower mold processing parameter (Fig. 12). Then, the corresponding upper mold parameters and other command information were added to the configuration file (Fig. 13).

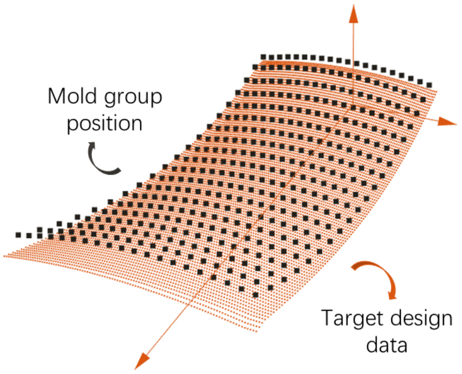


**Fig. 12 Regular points of machined surface to form the mold group.**

After the above four steps, we obtained the processing configuration file, which was consistent with the format required by our 3D CNC plate bending machines for ships. In addition, our method does not affect the use of existing CAD systems and 3D CNC plate bending machines in production. Our proposed approach can be integrated well into the existing production environment. Thus, the cost of updating the production system is very small.

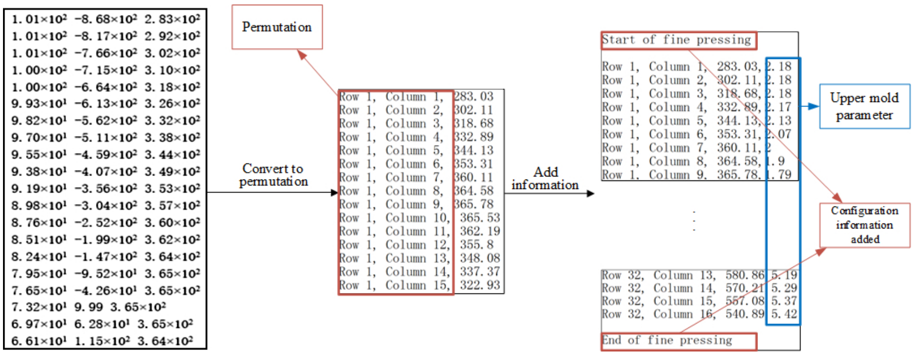
## 5.2 Experiment result

In this section, we conduct an experiment to compare the CBR-based springback calculation approach with the basic one to quantitatively study their performance.

The data set for testing is the ship plate data of six categories with 20 copies for each category. The case library constructed in Section 4.3 does not contain the testing data but those of similar cases. To evaluate the accuracy of both methods, we calculated the error rate for each ship plate with Eq. (6).

$$e=\sum_{i=1}^{k}\frac{\left|S_i-S_i'\right|}{S_i}\Big/k\times 100\% \tag{6}$$

where $S'$ is the prediction results, and $S$ is the actual springback. Then, we calculated the average value of the error rate by category. The experiment computes the


```
1. 01×10² −8. 68×10² 2. 83×10²
1. 01×10² −8. 17×10² 2. 92×10²
1. 01×10² −7. 66×10² 3. 02×10²
1. 00×10² −7. 15×10² 3. 10×10²
1. 00×10² −6. 64×10² 3. 18×10²
9. 93×10¹ −6. 13×10² 3. 26×10²
9. 82×10¹ −5. 62×10² 3. 32×10²
9. 70×10¹ −5. 11×10² 3. 38×10²
9. 55×10¹ −4. 59×10² 3. 44×10²
9. 38×10¹ −4. 07×10² 3. 49×10²
9. 19×10¹ −3. 56×10² 3. 53×10²
8. 98×10¹ −3. 04×10² 3. 57×10²
8. 76×10¹ −2. 52×10² 3. 60×10²
8. 51×10¹ −1. 99×10² 3. 62×10²
8. 24×10¹ −1. 47×10² 3. 64×10²
7. 95×10¹ −9. 52×10¹ 3. 65×10²
7. 65×10¹ −4. 26×10¹ 3. 65×10²
7. 32×10¹ 9. 99 3. 65×10²
6. 97×10¹ 6. 28×10¹ 3. 65×10²
6. 61×10¹ 1. 15×10² 3. 64×10²
```

```
Row 1, Column 1,  283. 03
Row 1, Column 2,  302. 11
Row 1, Column 3,  318. 68
Row 1, Column 4,  332. 89
Row 1, Column 5,  344. 13
Row 1, Column 6,  353. 31
Row 1, Column 7,  360. 11
Row 1, Column 8,  364. 58
Row 1, Column 9,  365. 78
Row 1, Column 10,  365. 53
Row 1, Column 11,  362. 19
Row 1, Column 12,  355. 8
Row 1, Column 13,  348. 08
Row 1, Column 14,  337. 37
Row 1, Column 15,  322. 93
```

```
Start of fine pressing

Row 1, Column 1,  283. 03, 2. 18
Row 1, Column 2,  302. 11, 2. 18
Row 1, Column 3,  318. 68, 2. 18
Row 1, Column 4,  332. 89, 2. 17
Row 1, Column 5,  344. 13, 2. 13
Row 1, Column 6,  353. 31, 2. 07
Row 1, Column 7,  360. 11, 2
Row 1, Column 8,  364. 58, 1. 9
Row 1, Column 9,  365. 78, 1. 79

        ⋮

Row 32, Column 13,  580. 86 5. 19
Row 32, Column 14,  570. 21 5. 29
Row 32, Column 15,  557. 08 5. 37
Row 32, Column 16,  540. 89 5. 42

End of fine pressing
```

**Fig. 13 Process parameters of the mold group adapted to equipment formation.**

average error rate of each category with the CBR-based springback computation approach (i.e., with CBR) and the one with the basic springback computation approach (i.e., without CBR). Table 1 and Fig. 14 present the experimental results.

**Table 1 Average error rate with and without CBR in different 3D model types.**

| 3D model type | Average error rate (%) | |
|---|---|---|
| | Without CBR | With CBR |
| Cylindrical | 5.086 63 | 2.997 85 |
| Conical | 20.806 30 | 16.855 35 |
| Sail | 23.955 33 | 22.158 55 |
| Saddle | 17.314 06 | 13.378 13 |
| Transverse bending and twisting | 9.977 32 | 8.126 98 |
| Transverse wave bending | 18.940 12 | 5.722 22 |

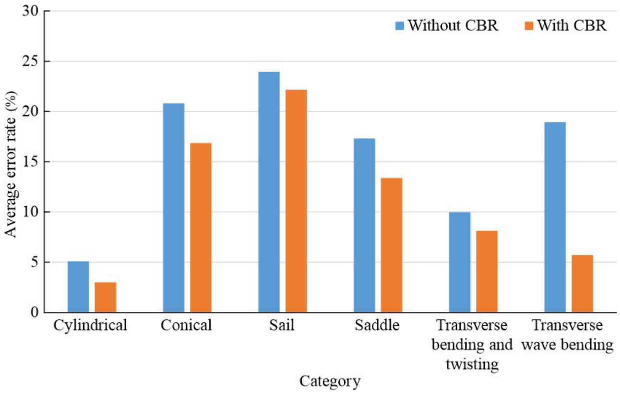


**Fig. 14 Comparison of average error rate with and without CBR preprocessing.**

Table 1 shows that the average error rate in all categories with CBR is lower than those without. Moreover, no evident relationship exists between the average error rate without CBR and its decline with CBR (Fig. 14). The experimental result indicates that CBR can improve the accuracy, and the optimization effect is independent of the improved DeepFit model performance. Therefore, CBR is effective and thus necessary in springback calculation.

## 6 Comparison and Discussion

We compare the methods described in this paper with the existing springback control approaches from eight aspects, including business needs, model representation, model matching and mapping, adaptability, expansibility, interpretability, and accuracy. Table 2 lists their differences and comparison results.

We first differentiated the approaches from their business needs, model representation, and modeled matching and mapping. The numerical simulation approach was used to predict springback and simulate the multi-point forming and unloading process of the hull-surface outer plate by constructing the finite element model. In this approach, the surface after springback was obtained, and the compensation was calculated by the normal correction method. The theoretical approach studied the springback compensation prediction for single curvature surfaces and used the theory of mechanics to deduct the curvature relationship of plate

**Table 2 Purpose, method, and performance comparison of related approaches.**

| Criteria | Numerical simulation[22] | Theoretical calculation[23] | Machine learning[24] | Hybrid[25, 26] | Our method |
|---|---|---|---|---|---|
| Business needs | Springback prediction | Springback compensation prediction for single curvature surface | 3D shape recognition, feature capture and classification | Springback prediction | Springback compensation prediction for multiple types of surfaces |
| Model representation | Finite element model | Discrete points and microelements | 2D projection view | Discrete points | Point cloud model |
| Model matching & mapping | Finite element method-based simulation | Curvature relationship deduction | Neural network | Neural network | CBR + Neural network |
| Adaptability | Low due to low model reusability | Low due to theoretical difficulty | High through model transformation | Medium due to highly demanded feature and sample size selection | High due to case reusability |
| Expansibility | Low from manual model creation | High from model reuse | High from model reuse | High from model reuse | High from model reuse |
| Interpretability | High | High | Low | Low | High |
| Accuracy | Medium due to simplified working condition | Low due to limited factors | Low due to information loss | Medium, affected by parameter types | High due to comprehensive information extraction |

springback based on the model of discrete points and microelements. The machine learning approach transforms CAD modeled into multi-angle images and uses the neural network of image processing to obtain the model characteristics. This approach was mostly used for classifying and recognizing CAD models. The hybrid approach combined theoretical calculation with machine learning results by considering the theoretical factors of springback as a neural network input to fit the springback based on the model of discrete points. These theoretical factors are usually obtained through sensitivity experiments.

From the perspective of adaptability, the numerical simulation needs to establish different finite element models for each plate when calculating the springback. Theoretical calculations can be used to obtain a better curvature surface. However, the construction of the quantitative relationship of complex surfaces, such as a hyperbolic surface, is difficult. In terms of feature extraction, different CAD models, which exhibit high adaptability, can be projected to form images for processing. The hybrid method requires different sensitivity experiments for various types of plates. However, in the case of the same plate type, after selecting the influencing factors as input, the hybrid method also exhibits high adaptability. The method proposed in this paper combines CBR and machine learning and constructs rules for transforming the CAD model into a unified point-cloud model. This method has high adaptability to ship plates with different shapes and design rules and 3D CNC plate bending machines with various processing rules.

From the perspective of expansibility, a new finite element model needs to be constructed every time in the theoretical calculation, indicating a low expansibility. After constructing the corresponding formula or network for a certain kind of ship plate surface, theoretical calculation, machine learning, hybrid method, and the method proposed in this paper can be used to process a large number of ship plate data efficiently and with high interpretability.

From the perspective of interpretability, numerical simulation and theoretical calculation are in white-box processing modes. When users use these two methods, they gain a clear understanding of the influence mechanism of springback and its strong interpretability. The network used by machine learning and hybrid methods is complex and with poor interpretability. This paper combines CBR to find similar cases to provide a basis for the results of machine learning fitting and enhance the interpretability of the method.

From the perspective of accuracy, although the numerical simulation simulates the whole process of multi-point pressure and unloading, it still simplifies the working conditions of processing. When calculating the springback of single curvature surfaces in the theoretical calculation, the influencing factors of springback are limited, the quantitative relationship is difficult to use when describing the springback process, and the accuracy of the final result is limited. Machine learning methods can only be used in the classification and recognition of CAD models. The accuracy of the hybrid method depends on the selected input factors and the number of data samples. However, many factors affect the springback, and the number of existing ship plates is small. Therefore, the accuracy still needs to be improved. In this paper, a point cloud is used to represent the ship plate surface, and the machine learning method for the point cloud is used to calculate springback. Considering the shape of the surface, combined with the CBR method, point-cloud can deal with the problem of small samples, ensure the reliability of results, and enhance the accuracy of machine learning methods.

Overall, numerical simulations and theoretical calculations are traditional springback computation approaches with strong theoretical support. However, given the complex factors of springback, these methods generally cannot meet the accuracy requirements and cannot be applied to complex surfaces. Machine learning and hybrid methods can achieve the mapping of nonlinear relationships better than neural networks. However, they are faced with the problem of incomplete feature selection and an insufficient number of samples, which result in the low accuracy of springback computation. By contrast, our approach combines machine learning with CBR as a comprehensive method for shape information extraction. This approach can ensure the accuracy and reliability of the method when the number of samples is small, as usually observed in ship plate processing.

## 7 Conclusion

In the process of 3D surface bending, the uncertainty of springback affects the efficiency and accuracy, resulting in the wastage of human and material resources. In this research, we proposed a springback computation approach based on the point cloud. This approach combines the CBR-based and machine learning methods for point-cloud model matching and springback prediction. To implement the approach, we devised the MMFBPG framework for automated 3D CNC plate bending-machine parameter generation. Based on the framework, we presented a case study on ship-saddle plate bending to show the feasibility of the solution. Meanwhile, we show the advantage of the proposed approach with CBR-based model matching by comparing it with an approach of deep learning model through experiments. The experimental results showed that the CBR-based approach can achieve a substantially higher computation accuracy than the machine learning-based approach. In summary, this work provides a new idea for the nonlinear mapping of 3D surfaces.

In future works, we will further study the influence on bending results based on the thickness, material, and other characteristics of 3D plates. In addition, our method can fully consider environmental factors, such as temperature and humidity to improve the accuracy and efficiency of 3D bending.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61972243).

## References

[1] Q. F. Zhang, Compensation algorithm for springback in multi-point forming and its validation by numerical simulation and experimental methods, (in Chinese), PhD dissertation, Jilin University, Changchun, China, 2014.

[2] X. D. Zhang, S. L. Liu, Y. Liu, X. F. Hu, and C. Gao, Review on development trend of launch and recovery technology for USV, (in Chinese), *Chin. J. Ship Res.*, vol. 13, no. 6, pp. 50–57, 2018.

[3] Y. X. Feng, Z. F. Zhang, G. D. Tian, Z. H. Lv, S. X. Tian, and H. F. Jia, Data-driven accurate design of variable blank holder force in sheet forming under interval uncertainty using sequential approximate multi-objective optimization, *Future Generat. Comp. Syst.*, vol. 86, pp. 1242–1250, 2018.

[4] Y. C. Gao, Z. X. Zhang, Y. X. Feng, M. Savchenko, I. Hagiwara, and H. Zheng, Flexible mesh morphing in sustainable design using data mining and mesh subdivision, *Future Generat. Comp. Syst.*, vol. 108, pp. 987–994, 2020.

[5] Z. J. Hou and L. G. Gu, Triangulation based on 3D reconstruction technique, *Appl. Mech. Mater.*, vol. 121–126, pp. 4249–4253, 2011.

[6] W. Wang, T. Y. Su, H. X. Liu, X. F. Li, Z. Jia, L. Zhou, Z. L. Song, and M. Ding, Surface reconstruction from unoriented point clouds by a new triangle selection strategy, *Comput. Graph.*, vol. 84, pp. 144–159, 2019.

[7] X. J. Qin, Z. T. Hu, H. B. Zheng, and M. Y. Zhang, Surface reconstruction from unorganized point clouds based on edge growing, *Adv. Manuf.*, vol. 7, no. 3, pp. 343–352, 2019.

[8] A. N. Ravari and H. D. Taghirad, Reconstruction of B-spline curves and surfaces by adaptive group testing, *Comput. Aided Des.*, vol. 74, pp. 32–44, 2016.

[9] W. J. Wang, L. L. Lai, J. Chen, and Q. Y. Wu, CAM-based non-local attention network for weakly supervised fire detection, *Serv. Oriented Comput. Appl.*, vol. 16, no. 2, pp. 133–142, 2022.

[10] W. Zhou, J. Y. Jia, C. X. Huang, and Y. Q. Cheng, Web3D learning framework for 3D shape retrieval based on hybrid convolutional neural networks, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 93–102, 2020.

[11] M. Gadelha, R. Wang, and S. Maji, Multiresolution tree networks for 3D point cloud processing, in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 105–122.

[12] J. J. Wu, C. K. Zhang, T. F. Xue, W. T. Freeman, and J. B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 82–90.

[13] C. R. Qi, H. Su, K. C. Mo, and L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, presented at the 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 77–85.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5105–5114.

[15] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, Attentional PointNet for 3D-object detection in point clouds, presented at the 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 2019, pp. 1297–1306.

[16] W. H. Zhou, J. G. Lu, and W. L. Yue, A new semantic segmentation method of point cloud based on PointNet and VoxelNet, presented at the 2019 Chinese Control and Decision Conf. (CCDC), Nanchang, China, 2019, pp. 803–808.

[17] Y. Wang, Y. B. Sun, Z. W. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, Dynamic graph CNN for learning on point clouds, *ACM Trans. Graph.*, vol. 38, no. 5, p. 146, 2019.

[18] P. Yuan, C. F. Wang, Y. Hu, J. X. Li, C. Y. Zhang, W. Huang, and F. Yu, Development of large plate bending machine for shipbuilding with three-dimensional numerical control, (in Chinese), *Shipbuilding of China*, vol. 55, no. 2, pp. 122–131, 2014.

[19] D. Saini, S. Kumar, and T. R. Gulati, NURBS-based geometric inverse reconstruction of free-form shapes, *J. King Saud Univ.-Comput. Inform. Sci.*, vol. 29, no. 1, pp. 116–133, 2017.

[20] Y. Ben-Shabat and S. Gould, DeepFit: 3D surface fitting via neural network weighted least squares, in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 20–34.

[21] D. W. Wei, H. S. Ning, F. F. Shi, Y. L. Wan, J. B. Xu, S. K.

Yang, and L. Zhu, Dataflow management in the Internet of Things: Sensing, control, and security, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 918–930, 2021.

[22] J. C. Wang, Y. J. Xie, R. C. Yang, J. C. Liu, and H. Zhou, Numerical analysis and process planning of cold bending and press forming of curved plate for offshore platform, *Ship Eng.*, vol. 43, no. 4, pp. 94–101, 2021.

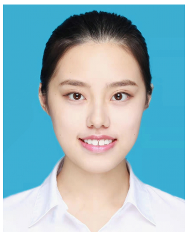
[23] Y. D. Chen, Numerical simulation study on the springback control of multi-point forming for curved hull plate, (in Chinese), Master dissertation, Jilin University, Changchun, China, 2015.

[24] Y. L. Fang, M. Xia, P. L. Ji, X. Zhou, M. Zeng, and X. G. Liu, Deep spherical panoramic representation for 3D shape recognition, (in Chinese), *J. Comput. Aided Des. Comput. Graph.*, vol. 29, no. 9, pp. 1689–1695, 2017.

[25] T. Q. Li, Research on springback prediction method of profile roll forming based on machine learning, (in Chinese), Master dissertation, Yanshan University, Qinhuangdao, China, 2019.

[26] Q. Y. Wei, Studies on the prediction of spring-back in air V-bending of sheet-metal, (in Chinese), Master dissertation, Jilin University, Changchun, China 2021.

**Min Zhu** received the BS degree in naval architecture from Shanghai Jiao Tong University, Shanghai, China, in 1998, and the MS degree from Shanghai Jiao Tong University, Shanghai, China, in 2010. He is currently pursuing the PhD degree at the School of Software, Shanghai Jiao Tong University, Shanghai, China. His research interests include industrial software, knowledge graph, big data, and machine learning.

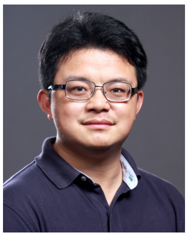


**Yanjun Dong** received the BS degree in software engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021, now she is pursuing the MS degree at the School of Software, Shanghai Jiao Tong University, Shanghai, China. Her research interests include industrial software and machine learning.

**Bingqing Shen** received the PhD degree in software engineering from University of Macau, Macau, China, in 2019, the MS degree in computer control and automation from Nanyang Technological University, Singapore, in 2008, and the BEng degree in communication engineering from Shanghai University, China, in 2006. He is a postdoctoral researcher at the School of Software, Shanghai Jiao Tong University, Shanghai, China. His research interests include distributed virtual world, digital twins, and blockchain technology.

**Haiyan Yu** received the PhD degree from Donghua University, Shanghai, China, in 2011. She has worked on integrating classical descriptive geometry to modern computing since 1998. She now is an associate professor at the School of Mechanical Engineering, Donghua University, Shanghai, China. She has published over 20 papers in the fields of geometry and graphics. She visited Stanford University in 2017 for the collaborative research in nano imaging. She is the council member of China Graphics Society and the secretary-general of the Committee of Graphics Computing. Her main research area is on computer graphics, geometric computing, and their applications in CAD and engineering modeling.

**Lihong Jiang** received the BS, MS, and PhD degrees from Tianjin University, Tianjin, China in 1989, 1992, and 1996, respectively. During 1992–1993, she worked as an assistant professor at the Department of Computer, Qingdao Ocean University, Qingdao, China. During 1996–1998, she worked as a postdoctoral researcher at the School of Management in Fudan University, Shanghai, China. She now is an associate professor at the School of Software, Shanghai Jiao Tong University, Shanghai, China. Her research interests include big data, knowledge graph, and machine learning.



**Hongming Cai** received the BS, MS, and PhD degrees from Northwestern Polytechnical University, Xi'an, China in 1996, 1999, and 2002, respectively. Now he is a professor at the School of Software, Shanghai Jiao Tong University, Shanghai, China. He is a standing director of China Graphics Society, a senior member of ACM/IEEE, and a senior member of China Computer Federation. He was rewarded as "National Outstanding Scientific and Technological Worker" by China Association for Science and Technology in 2012. His research interests include industrial software, computer graphics, geometric computing, knowledge graph, big data, and machine learning.