

# Image-based 3D Shape Reconstruction of Wind Turbine from Multiple Views

Minghao Huang<sup>1</sup>, Mingrui Zhao<sup>1</sup>, Yan Bai<sup>1</sup>, Renjie Gao<sup>1</sup>, Rongfeng Deng<sup>2,3</sup> and Hui Zhang<sup>1</sup>

<sup>1</sup>Programme of Computer Science and Technology, BNU-HKBU United International College, Zhuhai, 519087, China

<sup>2</sup>School of Industrial Automation, Beijing Institute of Technology, Zhuhai, 519088, China

<sup>3</sup>Centre for Efficiency and Performance Engineering, University of Huddersfield, Huddersfield, HD1 3DH, UK

n830026049@mail.uic.edu.cn (FA); amy Zhang@uic.edu.cn (CA);

## Abstract.

This paper addresses the problem of reconstructing depth and silhouette images of wind turbine from its photos of multiple views using deep learning approaches, which aims for wind turbine blade fault diagnosis. Some previous multi-view based methods have extracted each photo's silhouette and combined them into separate channels as the input of convolution; others use LSTM to combine a series of views for reconstruction. These approaches inevitably need a fixed number of views and the output result is divergent if the order of the input views is changed. So, we refer to a network, SiDeNet [3], which has a flexible number of input views and will not be affected by the input order. It integrates both viewpoint and image information from each view to learn a latent 3D shape representation and use it to predict the depth of wind turbine at input views. Also, this representation could generalize to the silhouette of unseen views. We make the following contributions to SiDeNet: improving the resolution of predicted images by deepening network structure, adopting 6D camera pose to increase the degrees of freedom of viewpoint to capture a wider range of views, optimizing the loss function of silhouette by applying weights on edge points, and implementing silhouette refinement with point-wise optimizing. Additionally, we conduct a set of prediction experiments and prove the network's generalization ability to unseen views. Evaluating predicted results on a realistic wind turbine dataset confirms the high performance of the network on both given views and unseen views.

**Keywords:** 3D representation, Multi-view reconstruction, Silhouette prediction, Depth prediction, Wind turbine dataset

## 1 Introduction

Wind energy is one of the most technologically mature and widely used renewable energy sources in the world. The number and usage of wind turbines has increased rapidly, and the frequency of accidents has also increased. At the same time, blade failures is the most expensive core component in wind turbines, accounting for more

than 20% of the total number of failures. In order to reduce the economic losses caused by this, new and effective wind turbine blade monitoring methods have become extremely important. Generally, sensors are installed on the wind turbine blade for fault diagnosis [22]. Due to the limitations and deficiencies of the sensor, the quality of the detection results mainly depends on the sensor itself and the use of the sensor, and the results often conflict. Therefore, the computer vision technology that does not need to use the sensor has received more attention. In order to obtain the 3D data of the wind turbine blade, researchers stick many markers on the wind turbine blade [23]. However, this method always takes a lot of time and requires higher image resolution and acquisition speed. Meanwhile, in order to diagnose the shape and status of wind turbine blades, we need to reconstruct it from a single or multiple photos.

This work proposes wind turbine blade (WTB) fault diagnosis based on a real-time 3D reconstruction system. In order to implement this system, a series of continuous views of the wind turbine must be generated to form a global 3D reconstruction of the wind turbine. Therefore, our task is to predict depth images of wind turbines from distinct perspectives. Due to the lack of prior knowledge, it is difficult to infer the concavities of the unseen views on the basis of single or multi view reconstruction. An alternative way to predict new unseen views is to use silhouette. Inspired by the deep learning network SiDeNet, we can use a variable number of views as input to predict the depth of a given view and the contour of a new view. The network can learn 3D information from single or multiple views in order to infer 3D shapes by combining the information of each view into global information. On this basis, SiDeNet can also be extended to new invisible shapes.

We also notice that a higher resolution of predicted images can increase the clarity of details. And by incorporating camera pose as 6 degree of freedom parameters, the method can be more universality applied. As our real dataset consists of several frames captured in a fixed camera viewpoint, rather than multiple-view data, single-view data, or data that focuses on only a small range of viewpoints is collected, we conducted experiments related to the number of views, and revealed the importance of the rich viewpoints in training data to compensate for the small number of views that we actually collect. Finally, in order to improve the edge accuracy of the predicted silhouette, we apply an improved silhouette weighted loss, and implement an optimized silhouette edge prediction method according to PointRend [1]. Then, a system that can generate a series of wind turbine views is well trained and evaluated.

Our work has the following contributions: 1) The encoder-decoder architecture and its loss function in SiDeNet are refined for learning 3D shapes and predicting depth and silhouette. 2) A dataset of complex wind turbine is synthesized, which shows that the learned 3D representation is sufficient for new view synthesis of a set of unseen objects with complex shapes and even textures. 3) By incorporating camera pose as 6 degree of freedom parameters, the method can be more universality applied.

This paper is organized as follows. Section 2 introduces basic knowledge of existing methods of several issues related to image-based reconstruction. Section 3 presents the main framework of SiDeNet and our improvements. Section 4 gives the description and analysis of proposed dataset. The results of conducted experiments are shown and evaluated in Section 5 which is followed by conclusions in Section 6.

## 2 Related Works

### 2.1 Sensors and Markers

The fault detection of fan blades mainly relied on the use of sensors [24] and markers [25]. For the sensor, the detection technology was mainly based on vibration signals [30]. However, under complex operating conditions, the signal is not sensitive to surface fault detection and is susceptible to environmental factors. Another detection technology based on acoustic emission technology requires densely arranged sensors on the surface of the blade [13]. This method is often limited by the acoustic emission sensor, and its orientation often affects the quality of the detection results.

Today's machine vision-based wind turbine blade detection can avoid the use of sensors, for example, Corten and Sabel [17] tried to use photogrammetry technology [18] to measure blades by presetting markers on wind turbine blades and towers. Ozbek [19] uses a system that includes 4 CCD cameras and a high-power flashlight to measure the working status of the fan through marked points. Poozesh [20] adds markers on the surface of the blade and uses a bunch of stereo cameras to capture its geometry. Moreno [21] uses a vision-based deep learning method to automatically monitor each part of the blade surface using a camera installed on a robotic system to detect damage. Combining the above methods and the existing literature, in the blade structure failure detection method, markers are added on the blade surface. However, these methods often have long detection cycles, high costs, and high camera resolution requirements.

### 2.2 3D reconstruction with multi-images

The single image method needs to apply a priori as a constraint to reconstruct 3D information, because the 3D shape of the model cannot be inferred from the feature correspondence between multiple images. For example, the prior may be class-based modeling for deviation from the average shape. This method was first proposed by Blanz and Vetter [15]. Another application is to use texture or lighting priors to recover various complex 3D shapes [14]. But this method that requires additional information as a priori is not suitable for our project.

The self-occlusion problem can be overcome by providing multiple viewpoints. Several methods to integrate information from different angles are proposed. A classic method is that given multiple views of an object, a 3D shape can be generated by combining the characteristic points of these views using structure-from-motion [16]. Our framework based on Wiles and Zisserman's SiDeNet [3] is optimized, using max-pooling to combine feature vectors extracted from multiple views into latent feature encoding, which can be used for the depth and silhouette prediction of each view.

### 2.3 Silhouette

The initial method [11] used a series of silhouette images with known camera extrinsics to predict the visual hull, which was achieved by using voxels in the 3D representation. This is an improvement over other traditional methods that take into account reconstruction shapes with certain geometric and photometric constraints [12]. When

testing constraints on shapes, they require multiple views and cannot infer an invisible view of the object. In our work, we construct a latent 3D representation by combining the encoding of multiple views. Given the input of the viewpoint information, the representation can be generalized to the silhouettes of the invisible part of the object.

## 2.4 Depth

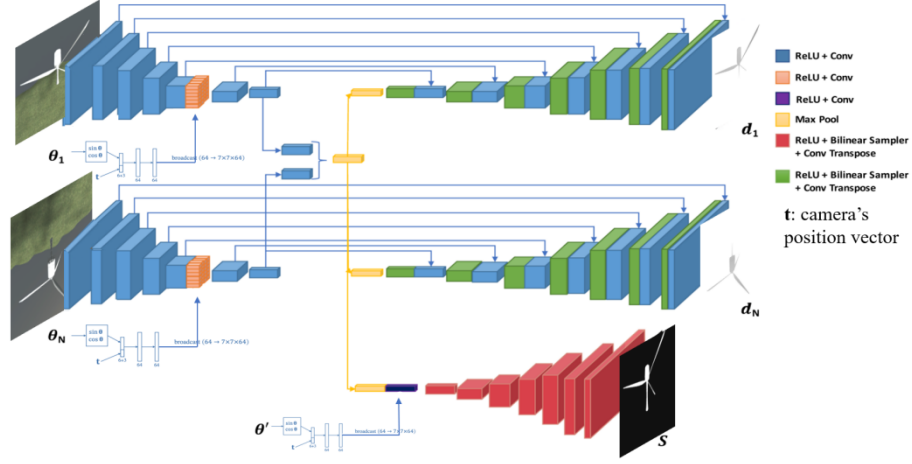
The traditional depth estimation method has the problem of matching errors in scenes with drastic changes in lighting conditions [9]. Yao [10] proposed an end-to-end depth estimation framework based on deep learning, which improves the accuracy of depth estimation by restoring the dense structure of scenes from multiple perspectives with a certain degree of overlap. Chen [8] used the predicted depth information combined with ground truth to form a three-dimensional point cloud, and then used the 3D point cloud algorithm to optimize the depth regression. However, since the generation of 3D point clouds requires a lot of resources, we adopt a more efficient method, which is based on U-Net [7] and Pix2Pix [6] using 2D images for depth recovery.

## 3 Proposed method

Inspired by the architecture of SideNet, we use a framework includes two branches, i.e., silhouette prediction and depth prediction at given viewpoints. The 6D pose [4] of the wind turbine refers to the translation and rotation of the camera coordinate system relative to the world coordinate system, including the three rotation angles of the three camera directions and the camera position vector ( $t=[x,y,z]$ ).

### 3.1 Main Architecture

The overall network structure is Encoder-Decoder, which is classical in U-Net, as shown in Fig. 1. The encoder is implemented using convolutional layers. The layer parameters and design are based on the encoder of the pix2pix architecture [6] and the U-Net architecture [7]. Given a set of images from each viewpoint, the encoder takes these images  $I_i$  and the corresponding viewpoint parameters, including translation and spatial rotation  $(I_i, t_i, R_i)$  as input, and then encodes them and broadcast over the feature channels. After convolution, the feature vector of each viewpoint is combined into a single latent vector  $x$ .  $x$  incorporates features from each viewpoint, including image and viewpoint parameters. Then through max pooling,  $x$  represents the most “confident” features of each viewpoint, and can be used for depth and silhouette prediction using the decoder. The decoder of each viewpoint depth prediction takes  $x$  as an input to recover the surface concavities of the object with depth  $d_1 \dots d_N$  in each given viewpoint, by using a transposed convolution and up-sampling layer with skip connections (taken from the corresponding input branch for feature reuse). The layer parameters also come from the pix2pix and U-Net. Similarly, the silhouette decoder uses  $x$  as input to predict the silhouette  $S$  at a new viewpoint  $(t_i, R_i)$ , with a feature channel broadcasting of new viewpoint information. The layers in the silhouette decoder are the same as those in the depth decoder without skip connections.

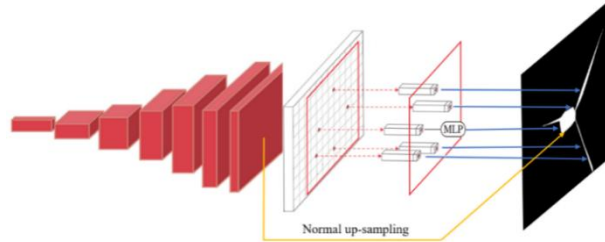


**Fig. 1.** The architecture of the network. Weights are shared across encoders. The blue arrows denote concatenation (over the feature channels). The feature vectors are combined to form a feature encoding (indicated by yellow blocks).

In order to have a more flexible depth estimation, we set the convolution encoder with camera pose parameters with 6 degrees of freedom into two full-connect layers [5]. The broadcasting operation is the same as before. Specifically, we first calculate the sin and cos of the 3 camera rotation angles, combine them with the camera position vector ( $t$ ), and pass them through two full-connect layers. The output is concatenated to each image encoder. These two layers can help reduce noise in camera poses and encourage the network to work well on each view. By incorporating the camera pose into the feature channel of the image encoder, we enable the network to learn information about the features of the wind turbine from specific views.

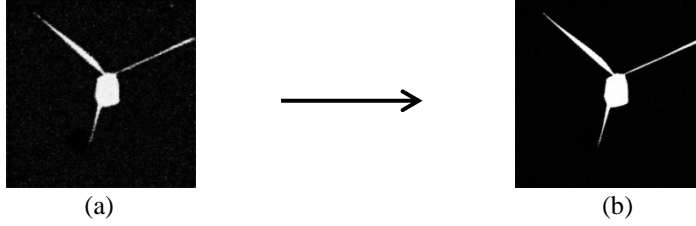
### 3.2 Silhouette Refinement

Silhouette estimation is always one of the most difficult steps in image segmentation. In order to obtain a good silhouette prediction, we adopt the method of PointRend [1], which effectively improves the accuracy of the silhouette. Basically, PointRend performs point-based segmentation prediction at adaptively selected locations based on an iterative subdivision algorithm.



**Fig. 2.** The coarse-to-fine rendering module of PointRend is a sub-branch between the last layer and output image where the corresponding points are refined and inserted. This achieves a point-wise refinement of uncertain feature points and improves the silhouette.

In Fig. 2, we add a coarse-to-fine rendering module between the last up-sampling layer and the output result on selected feature points, while original up-sampling remains unchanged. The module selects a set of points (red dots) from the output feature map according to the adaptive subdivision algorithm and makes prediction for each point independently with a small MLP [2] and then inserts these predicted results to the corresponding points of the output silhouette image. The result in Fig. 3 shows the improvement in silhouette edge clarity.



**Fig. 3.** Improvement of predicted silhouette. (a) Silhouette without point-wise refinement. (b) Silhouette with point-wise refinement. The result shows that the accuracy has improved.

### 3.3 Loss functions

This section introduces the use of multi-task loss functions, using binary cross-entropy loss to predict silhouettes and MSE loss to predict depth.

**Depth Loss.** The mean value of the difference between the target depth and the predicted depth is simply the absolute difference of all pixels of the target and predicted depth divided by the number of pixels. This operation allows the model no longer guess the absolute position of the object. It reduces ambiguity [3].

$$L_{depth} = \sum_{i=1}^N |d_i - d_{gt}| \quad (1)$$

where  $d$  is the predicted depth,  $d_{gt}$  is the ground truth of depth,  $N$  is the pixel number.

**Silhouette Loss.** Silhouette is a binary mask, i.e., 0 or 1. The binary cross-entropy is used to represent the average error of all pixels, which is also called the error prediction rate of all pixels. We use an improved loss: the use of parameter can weight more on the edge of silhouette, so that the silhouette edge can be predicted more accurately [3].

$$L_{sil} = \sum_{i,j} w_{i,j} (S_{i,j}^{gt} \log(S_{i,j}) + (1 - S_{i,j}^{gt}) \log(1 - S_{i,j})) \quad (2)$$

$$w_{i,j} = \begin{cases} dist_{i,j}, & \text{if } dist_{i,j} \leq T \\ c, & \text{otherwise} \end{cases}$$

where  $S$  is predicted silhouette,  $S^{gt}$  is ground truth of silhouette,  $i, j$  are position of pixel,  $c$  is default constant weight,  $T$  is default threshold of distance.

### 3.4 Interpretability

First, in the Encoder structure, the image is input to the convolutional layer of each encoder. In a certain layer, the corresponding viewpoint information composed of the 6D camera extrinsic is encoded and broadcast over the feature channels of each input convoluted image. Then through the convolutional layer, the broadcast feature channel of the viewpoint information is integrated into the original feature channels of the image. At the end of each encoder, a 512x1x1 feature vector is generated. All these

feature vectors are concatenated into  $512 \times N \times 1$  feature maps ( $N$  is the number of input viewpoints). Then through max pooling, the largest one is selected among the 512 mappings of size  $N \times 1$ , and the output is a  $512 \times 1 \times 1$  feature vector  $x$ .  $x$  incorporates the largest of the  $N$  elements of each feature map, which means that it contains some viewpoint and image information. Among them, each viewpoint of the 512 features is the most important, and contains the most confident features of each viewpoint. Therefore,  $x$  can encode the properties of 3D shapes useful for both depth prediction and silhouette prediction in new views.

Then for the decoder of depth branch, skip connections and up-sampling layers with transposed convolution are used for depth reconstruction. Up-sampling helps to restore the resolution and pixels of the image, and skip connections help to reuse the features of the image in the encoder and recover its information. For the silhouette branch,  $x$  is input together with a feature channel broadcasting of new viewpoint information, and starting from noise, the cost function is minimized by using binary cross-entropy loss. When input to the network, the reconstructed silhouette should give a feature vector  $x'$  that is the same as  $x$ . This is because among the viewpoint-dependent features in  $x$ , those who are confident in the viewpoints close to the new viewpoint will be given a larger weight, and then the parameters of these related units will be updated through back propagation, and the most prominent features in the new poses will be introduced slowly, and finally a reconstructed silhouette will be obtained.

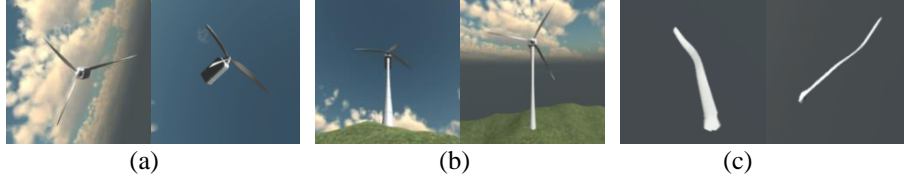
## 4 Dataset

Our work uses a series of wind turbine models to manually generate a data set. These models accurately simulate real wind turbines with realistic textures, including images and their corresponding pose information. Specifically, we align the center of the gravity and orientation of wind turbine itself with the world coordinate system ((0,0,0) represent the center coordinates), and then randomly select target points on the surface of a sphere with a radius of 2 centered at (0,0,0). Then images are generated with corresponding pose parameters.

As for rendering, the wind turbine model is normalized to a uniform size and 5~10 images of each wind turbine model are rendered from viewpoints uniformly and randomly selected within a specific range determined by the camera rotation angles and position vector, as the wind turbine is rotated about horizontal axis or vertical axis and has a variable distance from camera. In particular, the image sets of each model as data samples can cover the full-view of the wind turbine. These characteristics of our dataset help the network fully extract the information of various real turbine models from almost all possible views, and combine them to generate new views. In other words, the network parameters have been fitted to a wide range of views during training, so new views that may be close to the views already seen in training can be successfully generated in the test.

The dataset of wind turbine is divided into training set, validation set, and test set at the turbine level, and the proportions are 70%, 10%, and 20%, respectively. For each

iteration, wind turbines are randomly selected, from which a subset of the 5-10 rendered views are selected.



**Fig. 4.** Example images rendered from the dataset. (a, b) The wind turbine dataset. (c) The wind turbine blade dataset. These examples show various wind turbine models with rich shapes and textures, which helps the network generalize to real datasets.

## 5 Experimental results

### 5.1 Improvement of Resolution

The result of depth/silhouette prediction is shown in Fig. 6. The prediction error of the results of different architectures of SiDeNet is shown in Table 1. The prediction accuracy is high enough because compared with the ground truth, the loss of both depth and silhouette accurately lies in the range from 0.05 to 0.1, which is low enough for L1 loss and binary cross entropy loss, as explained with Eq. (1) and Eq. (2). Therefore, the predicted images of depth/silhouette in Table. 2 show that with the loss smaller than 0.1, the depth will have very little error on each pixel’s depth value compared with ground truth, and also silhouette will have a clear visual hull that is nearly the same as the one in ground truth because even though a few edge points are wrong predicted, they were refined with PointRend introduced in Section 3.

**Table. 1.** The performance of different architectures

Model	Input/Output size	Improved loss	Degree of Freedom	Depth error	Silhouette error
SiDeNet 256 <sub>1dof</sub>	256 × 256	×	1	0.078	0.101
SiDeNet 256 <sub>1dof</sub>	256 × 256	√	1	0.077	0.096
SiDeNet 256 <sub>6dof</sub>	256 × 256	√	6	0.072	0.093
SiDeNet 1024 <sub>1dof</sub>	1024 × 1024	√	1	0.061	0.065
SiDeNet 1024 <sub>6dof</sub>	1024 × 1024	√	6	0.063	0.064

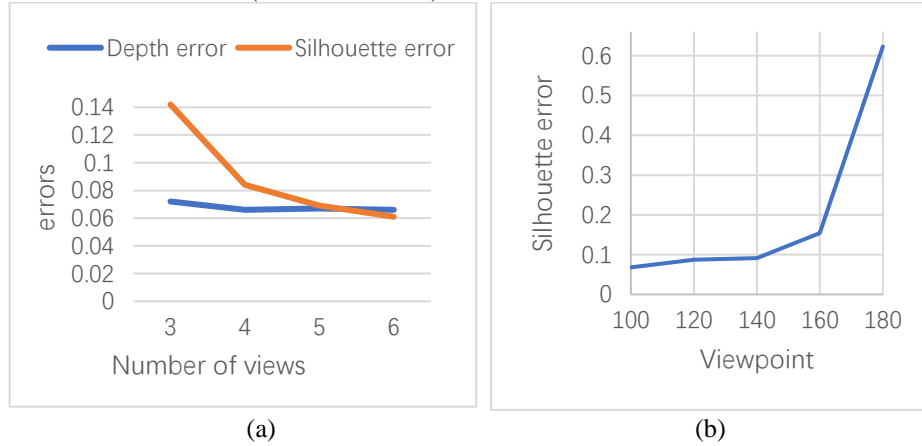
Table 1 compares the performance of SiDeNet 256 and SiDeNet 1024 with 1 or 6 degrees of freedom, as well as using an improved loss on silhouette prediction. This shows that although the difference in results driven by the increased degree of freedom is minimal, the converging speed during training is much faster because we have more feature variables related to the viewpoint, which helps the network encode the feature maps of multiple views more efficiently and perfectly. This helps to improve the robustness and compatibility of our multiple-view reconstruction task. In addition, along with the PointRend refinement, our weighted loss function improves



performance. In SiDeNet 1024, we increase the resolution of the input image from 256 to 1024. This can improve the resolution of predicted result. As a result, the accuracy of both depth and silhouette is increased.

## 5.2 The effect of the number of views

Training with more views can predict better than training with fewer views, and since the network sees more viewpoints during training, this can improve the prediction of those unseen views which are far from the range of predictable new views from input viewpoints. On this basis, when testing with a smaller number of views, the network can make use of information learned from the additional views. It shows that using fewer views than training will have certain negative impacts on the silhouette of unseen new views, and have no effect on the depth of input views. Given the limitation of view number in real data set, we have to find the reduced number of views that will not greatly affect the prediction performance of silhouette. We completed this experiment by reducing the number of views in testing (6 views in initial training). The result is shown in Fig. 5 (a). Therefore, we know that an acceptable silhouette accuracy is 0.084 which is with 4 views (2 views reduced).





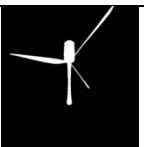



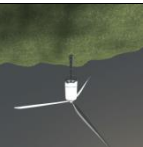

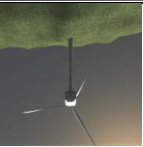

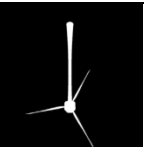
**Fig. 5.** (a) The effect of view number in testing on performance. (b) The performance of different viewpoint choices

## 5.3 The effect of viewpoint choice of generating new views

In order to test the viewpoint-encoding and generalization ability of SiDeNet, we control the viewpoint range of dataset by adjusting 6 degrees of freedom: only the horizontal rotation angle is changed in a range from  $0^\circ$  to  $120^\circ$ , and the other 5 degrees of freedom are fixed. The experimental results of silhouette prediction of different viewpoint choices are shown in Fig. 5 (b). It shows that the network is able to generate new views with high accuracy when the new viewpoint is selected near the viewpoints used in training (Viewpoints from  $100^\circ \sim 140^\circ$ ). This infers that we have to broaden the range of views and increase the number of views during training, so that we can train a network with stronger generalization ability.

#### 5.4 Predict Results

Based on the above experiment, we test the trained network using a data sample that contains 4 input views ( $0^\circ$ ,  $60^\circ$ ,  $120^\circ$ ,  $150^\circ$ ) and predict the depth images at these 4 given views as well as the silhouette images at new unseen views ( $45^\circ$ ,  $80^\circ$ ,  $180^\circ$ ) as shown in Fig. 6. which reveals that adding more views to the training will more accurately generate new views close to the given view.

Viewpoint	Input views	Depth/Silhouette
$0^\circ$		
$45^\circ$	Nil	
$60^\circ$		
$80^\circ$	Nil	
$120^\circ$		
$150^\circ$		
$180^\circ$	Nil	

**Fig. 6.** Predicted images of all available views ( $0^\circ \sim 180^\circ$ ) of a wind turbine model. For “Input views”, “Nil” means it is a new unseen viewpoint for silhouette prediction, otherwise there is an RGB image of the input view and this viewpoint is for depth prediction.

## 6 Conclusions

This paper introduces a multi-view 3D shape reconstruction system for wind turbine blade fault diagnosis. We make improvements on SiDeNet, in which the view-dependent encoders and feature combiner stimulates the network to integrate image information from all the input views. Therefore, the network can learn global feature that include the 3D shape features of all views, thereby being able to predict the depth of input views and generalize to the silhouette of unseen views. After verifying the prediction results, we obtain the following conclusions. Firstly, the increase in the degree of freedom of the viewpoint allows the network to take images of a wider viewpoint range as input, so that the shape information from more viewpoints can be learned, and the generalization ability of unseen views is improved. Secondly, the resolution increase makes the depth and silhouette images with more details. Third, the effectiveness of PointRend's adaptive points selection and point-wise refinement is remarkable. In addition, experiments have been conducted to prove that using more views in training, the network can use fewer views during testing and guarantee higher prediction performance. The experiments have also proved that the network more accurately predicts the silhouette of a new view that is close to the viewpoint of the views in training. These reflect the correlation and continuity between the image features of continuous viewpoints that the network can learn, which prove the generalization ability of the network to predict continuous views of the turbine. Finally, a series of consecutive views can be generated in high definition and combined into a reconstructed model.

**Acknowledgements** The work was supported by the National Natural Science Foundation of China (Project no. 62076029) and an internal funding from United International College.

## References

1. Kirillov, A., Wu, Y., He, K., & R Girshick. PointRend: Image Segmentation As Rendering. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. (2020).
2. Yan, X., Yang, J., Yumer, E., Guo, Y., & Lee, H. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In Advances in neural information processing systems. (2016).
3. Wiles, O., & Zisserman, A.. Learning to predict 3d surfaces of sculptures from single and multiple views. International Journal of Computer Vision. (2018).
4. Xiang, Y., Schmidt, T., Narayanan, V., & Fox, D.. Posecnn: a convolutional neural network for 6d object pose estimation in cluttered scenes. (2017).
5. Kok, F., J Charles, & Cipolla, R.. Footnet: an efficient convolutional network for multiview 3d foot reconstruction. Asian Conference on Computer Vision. (2020).
6. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2017).

7. Ronneberger, O., Fischer, P., & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the international conference on medical image computing and computer assisted intervention. (2015).
8. Chen, Rui & Han, Songfang & Xu, Jing & Su, Hao. Point-Based Multi-View Stereo Network. 1538-1547. 10.1109/ICCV.2019.00162. (2019).
9. Laina, I. , Rupprecht, C. , Belagiannis, V. , F Tombari, & Navab, N.. Deeper Depth Prediction with Fully Convolutional Residual Networks. Fourth International Conference on 3d Vision. IEEE. (2016).
10. Yao, Yao & Luo, Zixin & Li, Shiwei & Fang, Tian & Quan, Long. MVSNet: Depth Inference for Unstructured Multi-view Stereo. (2018).
11. Laurentini, A. The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2), 150–162. (1994).
12. K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3D reconstruction. volume 84, pages 80–96. Springer, (2009).
13. Qiu Zifeng. Research on Fan Blade Surface Damage Detection Based on Computer Vision [D]. Beijing Jiaotong University, (2019).
14. Barron, J., & Malik, J. Shape, illumination, and reflectance from shading. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37, 1670–1687. (2015).
15. Blanz, V., & Vetter, T. A morphable model for the synthesis of 3D faces. In Proceedings of the ACM SIGGRAPH conference on computer graphics (pp. 187–194). (1999).
16. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2 edn. Cambridge University Press, USA (2003).
17. Corten GP, Sabel JC. Optical motion analysis of wind turbines. In: Proceedings of European union wind energy conference. Goteborg, Sweden; 1996 May 20–24.
18. Johnson, J. T., Hughes, S., & van Dam, J. A stereo-videogrammetry system for monitoring wind turbine blade surfaces during structural testing. ASME Early Career Technical Journal , 8 (1), 1-1. (2009).
19. Ozbek, M., Rixen, D., Erne, O., & Sanow, G. Feasibility of monitoring large wind turbines using photogrammetry. Energy, 35(12), 4802-4811. (2010).
20. Poozesh, P., Baqersad, J., Niezrecki, C., Harvey, E., & Yarala, R. Full field inspection of a utility scale wind turbine blade using digital image correlation. CAMX, Orlando, FL , 10 (2.1), 2891-2960. (2014).
21. Vučina, D., Ćurković, M., & Novković, T. Classification of 3D shape deviation using feature recognition operating on parameterization control points. Computers in industry , 65 (6), 1018-1031. (2014).
22. Poozesh, P., Baqersad, J., Niezrecki, C., Avitabile, P., Harvey, E., & Yarala, R. Large-area photogrammetry based testing of wind turbine blades. Mechanical Systems and Signal Processing , 86 , 98-115. (2017).
23. Sarrafi, A., Mao, Z., Niezrecki, C., & Poozesh, P. Vibration-based damage detection in wind turbine blades using Phase-based Motion Estimation and motion magnification. Journal of Sound and vibration , 421 , 300-318. (2018).
24. Li Xiaowei. Research on fan blade detection based on fiber grating sensing technology [D]. Nanjing University of Aeronautics and Astronautics. (2013).
25. Wang, W., & Chen, A. Target-less approach of vibration measurement with virtual points constructed with cross ratios. Measurement , 151 , 107238. (2020).