3D Shape Reconstruction from Images in the Frequency Domain

Weichao Shen, Yunde Jia, and Yuwei Wu*
Beijing Laboratory of Intelligent Information Technology, School of Computer Science,
Beijing Institute of Technology, Beijing, 100081
{shenweichao,jiayunde,wuyuwei}@bit.edu.cn

Abstract

Reconstructing the high-resolution volumetric 3D shape from images is challenging due to the cubic growth of computational cost. In this paper, we propose a Fourier-based method that reconstructs a 3D shape from images in a 2D space by predicting slices in the frequency domain. According to the Fourier slice projection theorem, we introduce a thickness map to bridge the domain gap between images in the spatial domain and slices in the frequency domain. The thickness map is the 2D spatial projection of the 3D shape, which is easily predicted from the input image by a general convolutional neural network. Each slice in the frequency domain is the Fourier transform of the corresponding thickness map. All slices constitute a 3D descriptor and the 3D shape is the inverse Fourier transform of the descriptor. Using slices in the frequency domain, our method can transfer the 3D shape reconstruction from the 3D space into the 2D space, which significantly reduces the computational cost. The experiment results on the ShapeNet dataset demonstrate that our method achieves competitive reconstruction accuracy and computational efficiency compared with the state-of-the-art reconstruction methods.

1. Introduction

Deep neural networks have made good progress in 3D shape reconstruction owing to their powerful ability to extract priors from big data [18, 15, 21, 9, 6, 27, 12]. However, high-resolution 3D shape reconstruction is still challenging due to the cubic growth of computational cost. The high computational requirements may be reduced by using efficient data structures, such as prob [11] and Octree [22] in the spatial domain, but these methods often require more complicated training procedures and customized network architectures [19]. In this paper, we analyze the 3D shape in the frequency domain and propose a simple method to reconstruct the 3D shape in the 2D space with a general 2D convolutional neural network.

3D reconstruction in the frequency domain has been proven to be effective in medical and cryo-microscopy image processing [26, 25]. Much work [25, 24] shows that a 3D shape can be reconstructed from a series of 2D slices in the frequency domain. Considering that the surface of a 3D shape is always sparse, we assume that a volumetric 3D shape is reconstructed well with a compact set of the slices. Given the Fourier transform of a 3D shape at $32 \times 32 \times 32$ resolution, we select 2D slices along the coordinate axis direction, *i.e.*, from the low-frequency to the high-frequency. Figure 1 shows the inverse Fourier transform results. We found that the reconstruction error is reduced below 10% with only three slices selected along each axis direction. This observation motivates us to design a 3D shape reconstruction method in the 2D space by predicting slices in the frequency domain.

Different from the 3D shape reconstruction from medical images, in which each slice is calculated directly from a CT or MR image, we aim to train a model to predict slices from ordinary RGB or gray image. Due to the information gap between the spatial domain (images) and the frequency domain (slices), it is difficult to learn a projection function from an ordinary image to a slice using deep neural networks directly. To deal with this problem, we introduce an intermediate representation, thickness map. Our idea comes from the Fourier slice theorem (or Fourier projection-slice theorem), a famous theorem in medical image processing [1]. This theorem presents that a slice going through the origin of a 3D shape in the frequency space at an angle θ is equal to the Fourier transform of the 2D projection, which is the Radon transform of the 3D shape at the same angle θ. For the 2D slice without going through the origin, we extend the Fourier slice theorem and show that they are the Fourier transform of a weighted 2D projection, which is the Radon transform of the 3D shape after a sine-weighted preprocessing. Both the 2D projection and weighted 2D projection reflect the thickness of a 3D shape or weight 3D shape, so we name it the thickness map. A diagrammatic sketch can be found in Figure 2. The Radon transform projects a 3D

*corresponding author

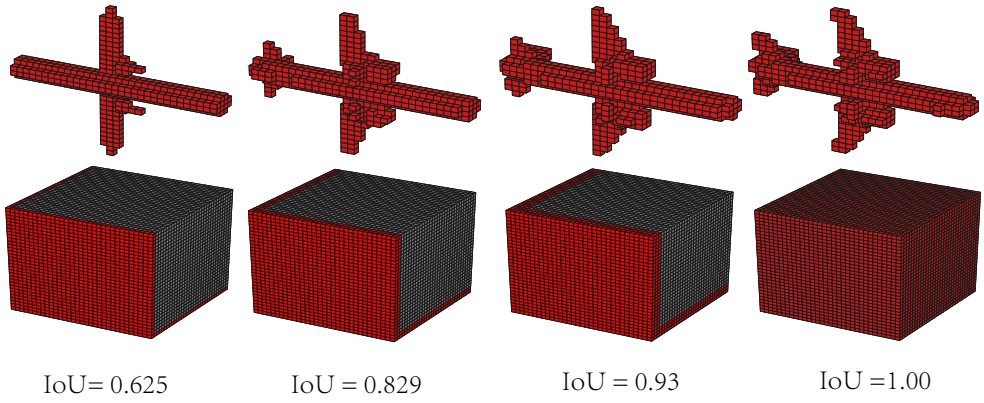

Figure 1. The reconstruction results of the 3D shape with the different number of 2D slices in the frequency domain. The bottom cubes are the Fourier transformation of $32 \times 32 \times 32$ 3D shape. One, two, and three red slices are selected along each axis, and the other pixels in gray are zero. The last one is the ground truth. The top shows the corresponding inverse Fourier transform results. The reconstruction accuracy reaches to 93% only with three selected slices along each axis direction.

shape into a 2D space in the spatial domain via line integral. Considering that the image is also a projection of the 3D shape in the spatial domain, the domain gap between the image and the thickness map is much smaller than that between the image and the slice, which benefits to learn a projection function with simple network architecture.

We leverage a deep neural network to predict thickness maps from images based on the auto-encoder architecture using simple 2D convolutional operation. Our network predicts the silhouette of the thickness map to learn the global information and predicts the edge of the thickness map to exploit the local details. The thickness map is generated by fusing the silhouette and edge using a sub-network. We apply the Fourier transform to thickness maps and embed them into the 3D space to get the Fourier transform of the 3D shape. The final voxel 3D shape is generated using the inverse Fourier transform. Our network requires a few computational resources as it only predicts a limited number of thickness maps with simple 2D convolutional layers. We evaluate our method on the ShapeNet dataset, and the experiment results show the effectiveness of our method.

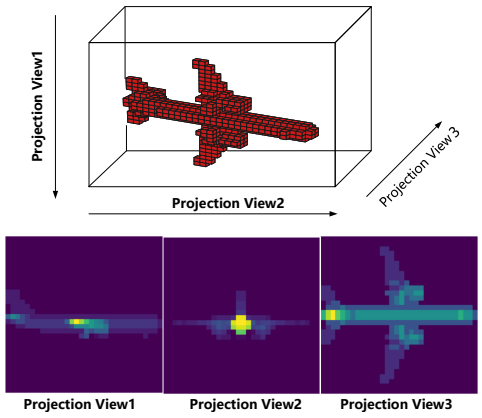

Figure 2. Example of thickness maps. We show three thickness maps of the 3D airplane model. The projection view is shown at the top. Each projection view is along a different axis of the Cartesian coordinate. The corresponding thickness map is shown at the bottom. The more lighten pixel means a more volumetric grid of the 3D shape belongs to the ray paralleling to the corresponding projection view.

In summary, the contributions of this work include:

- We propose a Fourier-based method for 3D shape reconstruction to reconstruct the 3D shape in the 2D space by predicting slices in the frequency domain, which can significantly reduce the computational cost.
- We introduce the thickness map to bridge the domain gap between the image in the spatial domain and the slice in the frequency domain. A deep neural network is built to generate the thickness map from the input images. Our network separately learns the global and local information by predicting the silhouette and edge of the thickness map, which can improve the reconstruction accuracy.

2. Related Work

High-resolution 3D volumetric grid reconstruction has been extensively studied for many years. Memory requirement of the high-resolution volumetric grid reconstruction approaches has been addressed with different data-adaptive discretization techniques, including Delaunay tetramerization [10], voxel block hashing [14], and Octree [3, 20, 17, 28]. To further get accurate reconstruction results, many recent high-resolution reconstruction methods were proposed

based on the deep neural networks. Tatarchenko *et al.* [22] proposed an Octree-based deep network for high-resolution 3D reconstruction. They designed new operations on the Octree, and the whole network can reconstruct the 3D shape in the efficient Octree space. Johnston *et al.* [7] reduced the complexity of the network by using a simple inverse discrete cosine transform layer replacing the original convolutional decoder. Hane *et al.* [5] designed a hierarchical surface prediction framework which reconstructs the high-resolution 3D shape via a coarse-to-fine strategy. Performing super-resolution on several orthographic depth projections, Smith *et al.* [19] reconstructed the 3D shape in the 2D space by up-sampling a low-resolution 3D shape. However, these methods reconstruct the 3D shape in the spatial space. In this paper, we propose to reconstruct a 3D shape by a compact set of 2D slices in the frequency domain. As a 3D shape can be represented using a few numbers of slices, our method can significantly reduce the computational cost.

The existing 3D reconstruction methods in Fourier domain were designed for special inputs, such as electron microscopy image [26], computed tomograph images [25], and striped lighting image [29]. Wang *et al.* [26] proposed a fast and accurate Fourier-based iterative reconstruction method that exploits the Toeplitz structure of the operator. Voropaev *et al.* [25] derived a Fourier-based reconstruction equation for computing laminography. Wu *et al.* [29] designed a two-variable 3D Fourier descriptors directly from a striped lighting system and proposed an iterative algorithm to compute the Fourier descriptors for both axisymmetric and nonaxisymmetric objects. All these methods in the frequency domain require specific image (CT and MRI) while our method can reconstruct a 3D shape just from ordinary RGB or gray images.

3. 3D Shape Reconstruction in the Frequency Domain

In this section, we provide the rigorous mathematical background of the thickness map and introduce the pipeline of the 3D shape reconstruction in the frequency domain.

3.1. Reconstruction Pipeline in the Frequency Domain

A 3D shape in the frequency domain is the Fourier transform of its spatial representation:

$$\begin{aligned}\mathscr{F}(O)(\omega_1,\omega_2,\omega_3) = &\int_{x_1,x_2,x_3} O(x_1,x_2,x_3) \\ &exp\big[-2\pi i(\omega_1 x_1+\omega_2 x_2+\omega_3 x_3)/N\big]dx_1x_2x_3,\end{aligned} \tag{1}$$

where O is a volumetric 3D shape in the spatial space and $\mathscr{F}(O)$ is the representation in the frequency space. N is the resolution of the 3D shape. $x_l|_{l=1,2,3}$ and $w_j|_{j=1,2,3}$ are the indices of the coordinate system in the spatial and the frequency space, respectively. Given $\mathscr{F}(O)$, we get the 3D shape in the spatial space by the inverse Fourier transform:

$$\begin{aligned}O(x_1,x_2,x_3) = &\int_{\omega_1,\omega_2,\omega_3} \mathscr{F}(O)(\omega_1,\omega_2,\omega_3) \\ &exp\big[2\pi i(\omega_1 x_1+\omega_2 x_2+\omega_3 x_3)/N\big]d\omega_1\omega_2\omega_3.\end{aligned} \tag{2}$$

As stated in the introduction, a volumetric 3D shape O can be reconstructed with a compact set of the slices of $\mathscr{F}(O)$. Referring to the attributes of the image in the frequency space, the reconstructed image with the low-frequency information maintains the global shape, and the high-frequency information adds local details [23]. Extending to 3D shape, we select a certain number of slices along the three axis orientations from the low-frequency to the high-frequency. Then, the 3D shape O can be reconstructed by

$$O = \mathscr{F}^{-1}\big(E[s^0_{\omega_1}, s^0_{\omega_2}, s^0_{\omega_3}; s^k_{\omega_j}; \ldots; s^n_{\omega_j}]\big), \tag{3}$$

where $s^k_{\omega_j}$ is the k-*th* slice along the axis ω_j ($j \in \{1,2,3\}$), $\mathscr{F}^{-1}$ is the inverse Fourier transform, and E is the artificial function to embed 2D slices into a 3D space. The total number of the slice selected along each axis is $n+1$.

3.2. Thickness Map

We will predict 2D slices $s^k_{\omega_j}$ from images. Predicting slices from the RGB image is difficult due to the domain gap between the image in the spatial domain and the slice in the frequency domain. To deal with this problem, we introduce a *thickness map* as the intermediate representation between an image and a slice. This representation is inspired by the Fourier slice projection theorem.

Theorem 1 (Fourier slice projection theorem [1])**.** *A slice going through the origin the Fourier transform of the 3D object at the orientation $\vec{r}$ equals to the Fourier transform of the corresponding projection image with the same orientation $\vec{r}$.*

The projection image P_r of the O at an orientation $\vec{r}$ is given by the Radon transform,

$$P_r = \int_{-\infty}^{\infty} O(x_1,x_2,x_3)dr. \tag{4}$$

We have three slices going through the origin of 3D shape, *i.e.*, ($s^0_{\omega_j}, j \in 1,2,3$). According to the Fourier slice projection theorem, these slices are the Fourier transform of the corresponding projection images.

$$\begin{aligned}s^0_{\omega_3} &= \mathscr{F}(O)(\omega_1,\omega_2,0) = \mathscr{F}(P^0_{\omega_3}), \\ s^0_{\omega_2} &= \mathscr{F}(O)(\omega_1,0,\omega_3) = \mathscr{F}(P^0_{\omega_2}), \\ s^0_{\omega_1} &= \mathscr{F}(O)(0,\omega_1,\omega_2) = \mathscr{F}(P^0_{\omega_1}),\end{aligned} \tag{5}$$

where

$$P^0_{\omega_j} = \int O(x_1,x_2,x_3)dx_j. \tag{6}$$

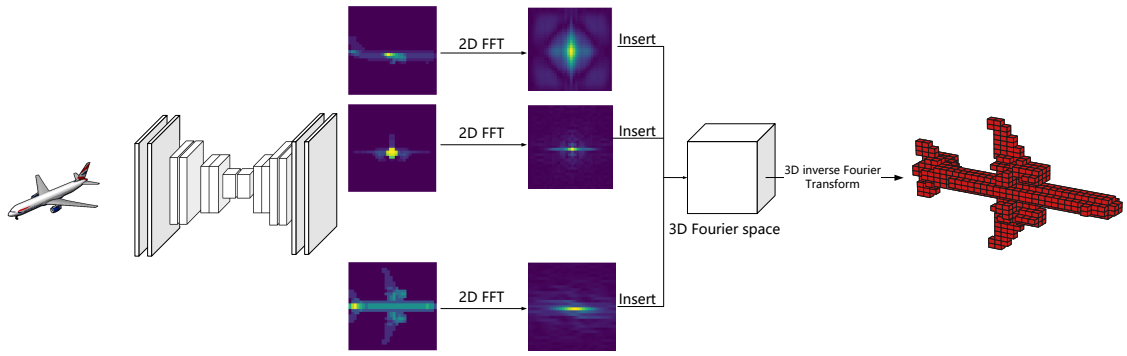

Figure 3. The pipeline of 3D reconstruction from images in the frequency domain. Taking an image as the input, we predict a series of thickness maps at different but specific projection views with a deep neural network. Each thickness map generates a 2D Fourier slice using the Fourier transform. All the 2D slices are inserted into a 3D Fourier space to constitute the Fourier transform of the 3D object. The 3D shape is reconstructed using the 3D inverse Fourier transform.

The Radon transform can be regarded as calculating the thickness of the 3D shape along a certain orientation, so that we name the projection image P as the thickness map. Considering that the Radon transform is the integral transform in the spatial domain, the gap between the thickness map and image is smaller than that between the slice and image. As a result, we propose to predict the thickness map from the image using deep neural networks and compute the slice from the predicted thickness map with the Fourier transform.

However, the Fourier slice projection theorem only provides the relationship between the thickness map $P^0_{\omega_j}$ and the slices $s^0_{\omega_j}$ through the origin of the 3D shape. We still need the other slices $\{s^k_{\omega_j}, k \neq 0\}$ to achieve a more accuracy reconstruction results. To deal with this problem, we extend the Fourier slice projection theorem and provide a way to calculate any slice $s^k_{\omega_j}$ from the corresponding thickness map.

Theorem 2. *The k-th slice $s^k_{\omega_j}$ along the axis ω_j is given by*

$$s^k_{\omega_j} = \mathscr{F}(P^k_{\omega_j}) = \mathscr{F}\big(R(P^k_{\omega_j})\big) - i\mathscr{F}\big(I(P^k_{\omega_j})\big), \tag{7}$$

where $R(P^k_{\omega_j})$ and $I(P^k_{\omega_j})$ are the Radon transform of the 3D shape O weighted by a cosine or sine function. i is the imaginary unit.

$$\begin{aligned} R(P^k_{\omega_j}) &= \int_{x_3} O(x_1, x_2, x_3) cos(-2\pi k x_j / N) dx_j, \\ I(P^k_{\omega_j}) &= \int_{x_3} O(x_1, x_2, x_3) sin(-2\pi k x_j / N) dx_j. \end{aligned} \tag{8}$$

Proof. When $j = 3$,

$$\begin{aligned} & s^k_{\omega_3} = \mathscr{F}(O)(\omega_1, \omega_2, k) = \int_{x_1,x_2,x_3} O(x_1, x_2, x_3) \\ & exp[-2\pi i(\omega_1 x_1 + \omega_2 x_2 + k x_3)/N] dx_1 x_2 x_3 \\ & = \mathscr{F}\Big\{ \int_{x_3} O(x_1, x_2, x_3) exp(-2\pi i k x_3 / N) dx_3) \Big\} \\ & = \mathscr{F}\Big\{ \int_{x_3} O(x_1, x_2, x_3) cos(-2\pi k x_3 / N) dx_3 \Big\} \\ & - i\mathscr{F}\Big\{ \int_{x_3} O(x_1, x_2, x_3) sin(-2\pi k x_3 / N) dx_3 \Big\} \\ & = \mathscr{F}(R(P^k_{\omega_3})) - i\mathscr{F}(I(P^k_{\omega_3})). \end{aligned} \tag{9}$$

Similarly, we can obtain Eq. (7) using the same way when $j = 1$ and $j = 2$. □

As shown in Eq. (8), the $R(P^k_{\omega_j})$ or $I(P^k_{\omega_j})$ is the Radon transform of the 3D shape O, so that predicting $P^k_{\omega_j}$ from the image has a smaller gap compared with predicting the slice. We still call $P^k_{\omega_j}$ as the thickness map. Referring to the Theorem 2, all the slices $s^k_{\omega_j}$ can be computed from the corresponding thickness map $P^k_{\omega_j}$. Now, our task is converted to determine the number of slices required for the accurate 3D reconstruction and predict the thickness maps from the input image with a deep neural network. The details can be found in Section 4 and Section 5.

Assuming that there is a trained network $\mathscr{N}$ that predicts the thickness maps from the image, a 3D shape is reconstructed from an input image I with Algorithm 1.

4. Sampling of the Thickness Map

Predicting more slices leads to a higher accuracy but a heavier computational burden. To make our model more efficient, we introduce a method to compute the high-frequency slices from the corresponding low-frequency slices.

Algorithm 1: 3D Reconstruction in the Frequency Domain

Input: Img: The input image.
$\mathscr{N}$: A trained thickness map prediction network.
Output: O: A reconstructed 3D shape.

Step 1*:* Put the image Img into the network $\mathscr{N}$ to predict a series of $R(P^k_{\omega_j})$ and $I(P^k_{\omega_j})$.
Step 2*:* Calculate the thickness map $P^k_{\omega_j}$ with $R(P^k_{\omega_j})$ and $I(P^k_{\omega_j})$ using Eq. (7).
Step 3*:* Calculate the slice $s^k_{\omega_j}$ using Eq. (5) and Eq. (7).
Step 4*:* Insert all slices $s^k_{\omega_j}$ into a 3D space at $w_j = k$ to generate the 3D shape $\mathscr{F}(O)$ in the frequency space.
Step 5*:* Apply the inverse Fourier transform on the $\mathscr{F}(O)$ to get the final 3D shape O using Eq. (3).
Return the reconstructed 3D shape O.

Theorem 3. *Given the k-th slice along the axis ω_j, where $j \in \{1, 2, 3\}$ and $k \neq 0$, the corresponding thickness map $P^k_{\omega_j}$ is writted as $P^k_{\omega_j} = R(P^k_{\omega_j}) + i \times I(P^k_{\omega_j})$. Then the slice at the high-frequency $\omega_j = N - k$ can be calculated by $P^{N-k}_{\omega_j} = R(P^k_{\omega_j}) - i \times I(P^k_{\omega_j})$. i is the imagary unit and N is the resolution of the 3D shape.*

Proof. When $j = 3$, the thickness map $P^k_{\omega_3}$ can be noted as

$$\begin{aligned} P^k_{\omega_3} &= R(P^k_{\omega_3}) + i \times I(P^k_{\omega_3}) \\ &= \int_{x_3} O(x_1, x_2, x_3) cos(-2\pi k x_3/N) dx_3 \\ &+ i \int_{x_3} O(x_1, x_2, x_3) sin(-2\pi k x_3/N) dx_3. \end{aligned} \tag{10}$$

For the $P^{N-k}_{\omega_3}$,

$$\begin{aligned} P^{N-k}_{\omega_3} &= R(P^{N-k}_{\omega_3}) + i \times I(P^{N-k}_{\omega_3}) \\ &= \int_{x_3} O(x_1, x_2, x_3) cos(2\pi k x_3/N - 2\pi x_3) dx_3 \\ &+ i \int_{x_3} O(x_1, x_2, x_3) sin(2\pi k x_3/N + 2\pi x_3) dx_3 \\ &= R(P^k_{\omega_j}) - i \times I(P^k_{\omega_j}). \end{aligned} \tag{11}$$

Similarly, we can achieve the same conclusion when $j = 1$ and $j = 2$. □

Based on Theorem 3, we predict $n + 1$ slices at the low-frequency and calculate $n + 1$ slices at the high-frequency, which can reduce half of the computational cost. Using this setting, we test the reconstruction accuracy of the 3D shape at a different resolution with a different number of slices, as shown in Table 1. The high-resolution 3D shape more slices to achieve a high accuracy. We select 3 slices for $32^3, 64^3, 128^3$ and 256^3 shape reconstruction.

Table 1. The reconstruction accuracy of the 3D shape at the different resolution with a different number of slices. "-" means that it is unnecessary to exam the reconstruction accuracy with more slices.

	Number of slices				
Resolution	1	2	3	4	5
32	0.721	0.942	0.987	-	-
64	0.783	0.932	0.972	0.983	-
128	0.841	0.941	0.969	0.980	-
256	0.787	0.913	0.955	0.968	0.973

5. Predicting the Thickness Map from Images

In this section, we build a deep neural network to predict the thickness maps from the input image.

5.1. Network Architecture

A thickness map should be bimodal. The global structure continuities create the shape of the 3D object, and local discontinuities describe the texture and details. Directly predicting the thickness map from the input image will output a good shape but lack the fine details, as minimizing the mean squared error results in blurry images without sharp edges [19]. To address this issue, we attempt to reconstruct a thickness map by predicting its silhouette and edge separately. The silhouette presents the global shape structure of the 3D shape, and the edge presents the fine local details. Separately predicting the global shape and fine details with different networks can reduce the complexity of the original learning problem, leading to an accurate result and avoiding the overfitting problem.

The silhouette S_p of the thickness map is predicted by a deep auto-encoder f_{sil}, *i.e.*, $S_p = f_{sil}(I)$. The encoder is a 2D convolution network, and the decoder is a 2D deconvolution network. The encoder consists of five convolutional layers with the group-normalization [30] and ReLU [13] activation. A fully connected layer converts the feature map into the vector with size 512. The decoder consists of *h* deconvolutional layers with stride size 2, where *h* changes with the resolution of 3D shape. A sigmoid activation function is applied to guarantee the network outputting the occupancy probability of each pixel.

The other deep neural network f_{edge} takes the image as input and predicts the edge map E_p of the thickness map, which presents the probability of each pixel of the thickness map belonging to an edge, *i.e.*, $E_p = f_{edge}(I)$. The architecture of the f_{edge} is similar to the f_{sil}, except that the kernel size of both the convolutional filter and deconvolutional filter is smaller to exploit the local information.

The outputs of the edge and silhouette network are combined together to generate the $R(\bar{P})$ and $I(\bar{P})$ for the thickness map $\bar{P}$ using a fully convolutional deep network f_{comb}, $\big[R(\bar{P}), I(\bar{P})\big] = f_{comb}(I, S_p, E_p)$. All the inputs are com-

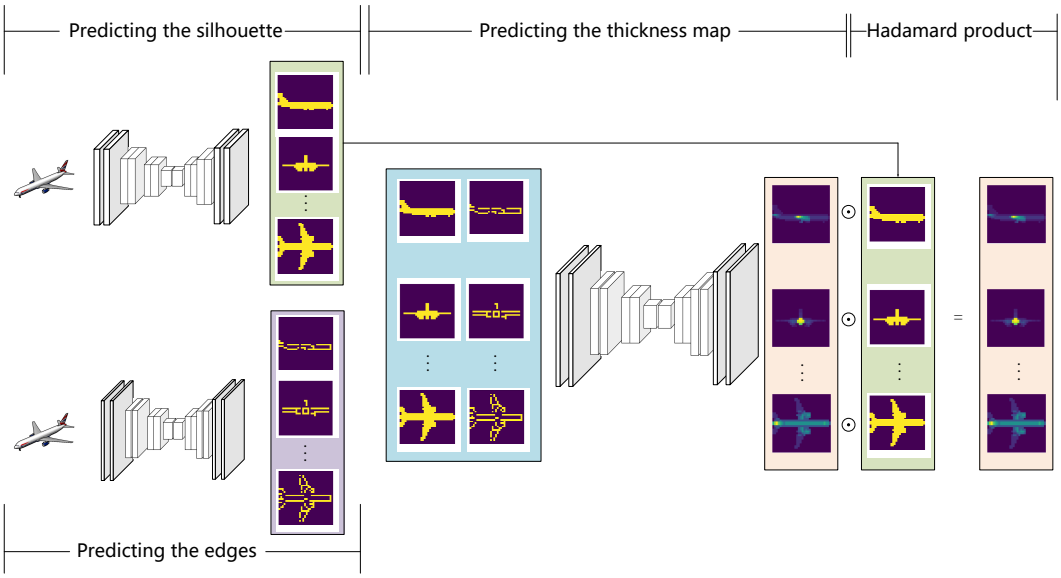

Figure 4. The architecture of the thickness map prediction network. Given an image, we predict a series of silhouettes and edges of the thickness map. All these silhouettes and edges are combined together to predict the thickness map.

bined together at channel dimension. f_{comb} is a bottleneck network with 7 blocks as the encoder and h blocks as the decoder, where h changes with the resolution of 3D shape. Each block in the encoder (or decoder) is composed of a convolutional (or deconvolutional) layer with group-normalization and ReLU activation. The activation of the last layer is a sigmoid function to guarantee that the output is the occupancy probability. At the end of f_{comb}, we can mask the output with the silhouette to avoid the noise out of the 3D shape.

$$P = \bar{P} \odot S_p, \tag{12}$$

where $\odot$ is the Hadamard product and P is the final predicted thickness map. However, this is not a necessary operation in our network. The whole architecture can be found in Figure 4.

5.2. Loss Functions

Three loss functions are introduced to train f_{sil}, f_{edge} and f_{comb}. The f_{sil} is trained by minimizing the cross-entropy between the predicted and truth silhouette of the thickness map,

$$\mathscr{L}_{sil} = \sum_{i=1}^{L} P_{sil} log(\hat{P}_{sil}) + (1 - P_{sil}) log(1 - \hat{P}_{sil}), \tag{13}$$

where $\hat{P}_{sil}$ is the ground truth silhouette which can be calculated from the 3D ground truth.

The loss function for f_{edge} is the square mean error between the predicted edge of thickness map and the ground truth,

$$\mathscr{L}_{edge} = ||\hat{P}_{edge} - P_{edge}||_2^2 + |\hat{P}_{edge} - P_{edge}|. \tag{14}$$

We add the ℓ_1 constraint into the loss function to release the loss of the fine details caused by the ℓ_2 constraint.

The last loss function for the whole network is the square mean error between the predicted thickness map and the ground truth,

$$\mathscr{L}_{thickness} = ||\hat{P} - P||_2^2. \tag{15}$$

$\hat{P}$ is the ground truth which is computed from the 3D shape using Eq. (6) and Eq. (8).

There is no consistency between different predicted thickness maps, so that the reconstructed 3D shape may have a low accuracy while each thickness map achieves a small reconstruction error. To deal with this problem, we use the 3D reconstruction loss $\mathscr{L}_O$ which is defined as

$$\mathscr{L}_O = \left|\left|\hat{O} - \mathscr{F}^{-1}\Big(E\big(\mathscr{F}(P)\big)\Big)\right|\right|_2^2. \tag{16}$$

$\hat{O}$ is the ground truth. $\mathscr{F}^{-1}\Big(E\big(\mathscr{F}(P)\big)\Big)$ is the reconstructed 3D shape using the predicted thickness maps P. This loss keeps the global geometry smooth of the reconstructed 3D shape. Furthermore, considering that the surface is sparse, the number of the voxel with a value 1 is much smaller than voxel with 0, so that Eq.(16) easily makes the network pay more attention to the blank regions of the 3D shape. Therefore, we separately calculate the errors on 3D shape surface and blank region. The new loss is designed as

$$\begin{aligned}\mathscr{L}_O = &\left|\left|\hat{O} - \hat{O} \cdot \mathscr{F}^{-1}\Big(E\big(\mathscr{F}(P)\big)\Big)\right|\right|_2^2 \Big/ \sum \hat{O} \\ &+ \left|\left|(1 - \hat{O}) \cdot \mathscr{F}^{-1}\Big(E\big(\mathscr{F}(P)\big)\Big)\right|\right|_2^2 \Big/ \sum (1 - \hat{O}).\end{aligned} \tag{17}$$

The final loss function is the simple linear combination of all items

$$\mathscr{L}_{final} = \mathscr{L}_{sil} + \mathscr{L}_{edge} + \mathscr{L}_{thickness} + \mathscr{L}_{O}. \quad (18)$$

6. Experiments

In this section, we evaluate our reconstruction method from both the computational efficiency and reconstruction accuracy. Both the memory cost and iterate time of the network are presented and compared with the popular high-resolution 3D reconstruction methods in Section 6.3. We show the reconstruction results on both the high-resolution and low-resolution 3D shape reconstruction in Section 6.4.

6.1. Dataset

We use the synthetic dataset ShapeNet [2] to evaluate our method. ShapeNet is a large 3D dataset with manually verified category and alignment annotations. Among all the 55 categories, we select a subset for our evaluation, *i.e.*, **ShapeNet-all**. ShapeNet-all is introduced by Choy *et al.* [4], which contains approximately $50,000$ CAD models from 13 main categories of the ShapeNet dataset. All data was voxelized in multiple resolutions using the *binvox* tool [16].

6.2. Experimental Setup

The network is trained using adaptive moment estimation [8] (Adam) with initial learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We decrease the learning rate by a factor of 0.5 after every 5K iterations. The training process is divided into two stages. We train f_{sil} and f_{edge} in the first 15 epochs. Then the whole network is trained with the final loss.

For the quantitative evaluation, we compute the Intersection over Union (IoU) measure between the ground truth and the predicted value. In general, a high IoU value means an accurate reconstruction result.

6.3. Computational Efficiency

In this section, we show the computational efficiency of our method and compare it with the OGN and the dense auto-encoder introduced in [22]. We compare the runtime and memory cost of all the methods at different resolutions. Our method is performed on an NVidia 1080 Maxwell GPU, with 12GB of memory. The batch size is set to 1 so that all networks can run at the largest possible resolution. Table 2 and Table 3 show the final results. Both of our method with and without $\mathscr{L}_O$ loss are reported.

At the low-resolution scenario, all the models have similar low computational cost, both in memory cost and iterate time. However, as the resolution grows, the OGN and our method is drastically faster and consumes far less memory cost. The memory cost of the dense network increases cubically due to the number of the dense voxel grid is determined by its cube resolution. As all the feature maps are 2D, the memory cost of our method without $\mathscr{L}_O$ increases squarely following the rise of the resolution, so that it achieves a comparable memory cost compared with OGN. Our method with $\mathscr{L}_O$ has a large memory because calculating the $\mathscr{L}_O$ requires the 3D high-resolution ground truth. Both of our network with $\mathscr{L}_O$ and without $\mathscr{L}_O$ achieve a faster iteration time compared with OGN.

Table 2. Memory cost of our method, OGN and a dense network at different output resolutions. Batch size is set to 1. (Ours + $\mathscr{L}_O$) is our method with $\mathscr{L}_O$ loss and (Ours - $\mathscr{L}_O$) is without $\mathscr{L}_O$.

	Memory,GB			
Resolution	Dense	OGN [22]	Ours - $\mathscr{L}_O$	Ours + $\mathscr{L}_O$
64^3	0.51	0.36	0.38	0.45
128^3	1.60	0.45	0.54	1.1
256^3	9.7	0.54	0.86	1.93
512^3	(74.28)	0.88	1.20	2.51

Table 3. Iteration time of our method, OGN and a dense network at different output resolutions. Batch size is set to 1. (Ours + $\mathscr{L}_O$) is our method with $\mathscr{L}_O$ loss function and (Ours - $\mathscr{L}_O$) is without $\mathscr{L}_O$.

	Iteration time, s			
Resolution	Dense	OGN [22]	Ours - $\mathscr{L}_O$	Ours + $\mathscr{L}_O$
64^3	0.21	0.06	0.034	0.04
128^3	0.63	0.18	0.13	0.15
256^3	3.22	0.64	0.23	0.47
512^3	(41.3)	2.06	0.86	1.21

6.4. Reconstruction Accuracy

In this section, we test our method on 3D reconstruction from a single image at the high-resolution $256 \times 256 \times 256$. Three categories in the ShapeNet-All (*i.e.*, "Cars", "Airplane" and "Chair") are selected considering that these categories contain enough training samples and have more shape variations. For each category, 80% of the data is used for training and 20% for testing. We compare our method with the popular high-resolution 3D reconstruction methods including Octree generation network (OGN) [10] and multi-view decomposition network (MVD) [19]. The experiment results are shown in Table 4. Our method achieves a comparable reconstruction accuracy compared with the state-of-the-art method MVD. Compared with the MVD, our method is more simple which do not need a complex space caving process. The output 3D shape can be easily generated with the inverse Fourier transform.

Our method utilizes partial information of the Fourier transform to achieve a balance between reconstruction accuracy and computational efficiency. To examine how this

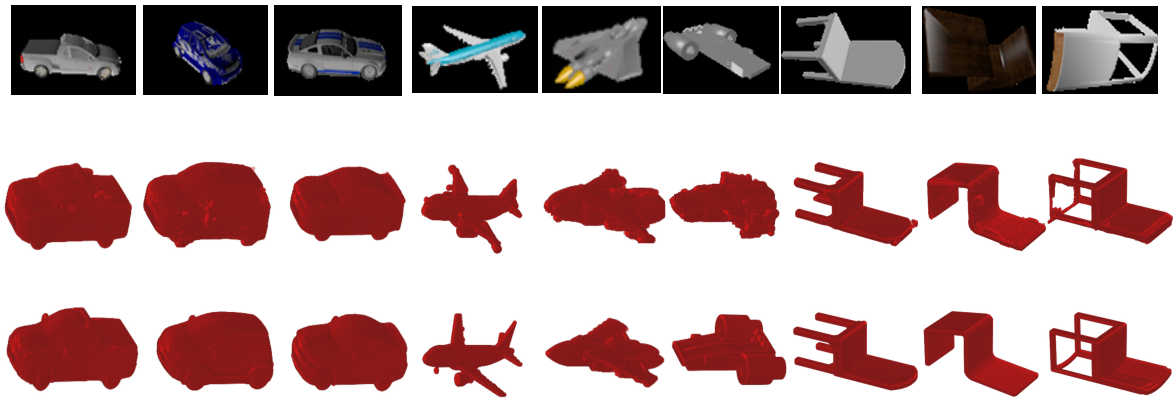

Figure 5. 3D shape reconstruction results (middle) of three classes in ShapeNet, *Car, Plane, and Chair* from the input image (top). The bottom is the ground truth.

Table 4. Single-image 3D reconstruction results on *car, airplane* and *chair* at resolution 256^3. "-" indicates that no result is reported in the original paper.

Category	OGN [22]	MVD [19]	Ours
Car	0.782	0.807	0.791
Airplane	-	0.589	0.581
Chair	-	0.433	0.425

Table 5. Single-view 3D reconstruction results on the 32^3 version of ShapeNet-all dataset. Our method achieves the best reconstruction result on most categories.

Category	R2N2 [4]	OGN [22]	Dense	Ours
Plane	0.513	0.587	0.570	**0.597**
Bench	0.421	0.481	0.481	**0.503**
Cabinet	0.716	0.729	**0.747**	0.726
Car	0.798	0.816	0.828	**0.841**
Chair	0.466	0.483	0.481	**0.521**
Monitor	0.468	0.502	0.509	**0.530**
Lamp	0.381	0.398	0.371	**0.421**
Speaker	0.662	0.637	0.650	**0.709**
Firearm	0.544	**0.593**	0.571	0.487
Couch	0.528	0.646	**0.668**	0.665
table	0.513	0.536	**0.545**	0.543
Cellphone	0.661	0.702	0.698	**0.713**
Watercraft	0.513	**0.632**	0.550	0.612
Mean	0.560	0.596	0.590	**0.605**

strategy influence the reconstruction accuracy, we introduced the low-resolution reconstruction experiment to compare our method with the baseline, a dense network introduced in [4]. We present the reconsturction results on all 13 classes of the ShapeNet-all dataset at the resolution $32\times32\times32$. We also compare our method with the other 3D reconstruction networks, including an auto-encoder LSTM based network (R2N2) [4] and OGN [10]. Table 5 shows the IoU result of all models. Our method achieves the best reconstruction results on most of the categories. The satisfactory performance on all classes presents the good generalization of our method.

To qualitatively evaluate the performance of our method, we show some reconstruction samples from the *"Cars"*, *"Airplane"* and *"Chiar"* class at resolution $256\times256\times256$. Figure 5 presents the reconstruction results.

7. Conclusion

In this paper, we have proposed a 3D shape reconstruction method from images in the frequency domain. We shown that a 3D shape can be reconstructed by a compact 2D slice set at a high reconstruction accuracy. We have exploited the Fourier projection slice theorem and introduced the 2D thickness map which can reduce the domain gap between the input image and 2D slices. A deep network was built to predict the thickness map from the input image by exploiting the edge and silhouette constraints. This network allows us to predict fine details (edges) and global shape (silhouettes) of thickness map separately from the input image, which allows a more accuracy reconstruction result. Using slices in the frequency domain, our method transferred the 3D shape reconstruction from the 3D space into the 2D space, which significantly reduces the computational cost. The experimental results on ShapeNet dataset validated that the proposed method can achieve satisfactory results in an efficient way.

8. Acknowledgement

This work was supported by the Natural Science Foundation of China (NSFC) under Grant No. 61773062.

References

[1] R. N. Bracewell. Strip Integration in Radio Astronomy. *Australian Journal of Physics*, 9:198, June 1956.

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[3] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(4):113, 2013.

[4] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[5] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3d object reconstruction. In *3D Vision (3DV), 2017 International Conference on*, pages 412–420. IEEE, 2017.

[6] L. Jiang, S. Shi, X. Qi, and J. Jia. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[7] A. Johnston, R. Garg, G. Carneiro, I. Reid, and A. vd Hengel. Scaling cnns for high resolution volumetric reconstruction from a single image. In *ICCV Workshops*, 2017.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *international conference on learning representations*, 2015.

[9] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[11] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas. Fpnn: Field probing neural networks for 3d data. In *NIPS*, 2016.

[12] Y. Liao, S. Donn, and A. Geiger. Deep marching cubes: Learning explicit surface representations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[14] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013.

[15] C. Niu, J. Li, and K. Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] F. S. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003.

[17] G. Riegler, A. Osman Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] D. Shin, C. C. Fowlkes, and D. Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] E. Smith, S. Fujimoto, and D. Meger. Multi-View Silhouette and Depth Decomposition for High Resolution 3D Object Representation. *ArXiv e-prints*, Feb. 2018.

[20] F. Steinbrücker, J. Sturm, and D. Cremers. Volumetric 3d mapping in real-time on a cpu. In *ICRA*, pages 2021–2028, 2014.

[21] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] J. Todd. Digital image processing (second edition): By r. c. gonzalez and p. wintz, addison-wesley, 1987. 503 pp. price: 29.95. (isbn 0-201-11026-1). *Optics and Lasers in Engineering*, 8(1):70–71, 1988.

[24] M. van Heel, B. E. Gowen, R. Matadeen, E. V. Orlova, R. A. Finn, T. Pape, D. Cohen, H.-U. Stark, R. Schmidt, M. Schatz, and A. Patwardhan. Single-particle electron cryomicroscopy: towards atomic resolution. *Quarterly reviews of biophysics*, 33 4:307–69, 2000.

[25] A. Voropaev, A. Myagotin, L. Helfen, and T. Baumbach. Direct fourier inversion reconstruction algorithm for computed laminography. *IEEE Transactions on Image Processing*, 25(5):2368–2378, May 2016.

[26] L. Wang, Y. Shkolnisky, and A. Singer. A Fourier-based Approach for Iterative 3D Reconstruction from Cryo-EM Images. *ArXiv e-prints*, July 2013.

[27] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4):72:1–72:11, July 2017.

[29] M. . Wu and H. . Sheu. Representation of 3d surfaces by two-variable fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):858–863, Aug 1998.

[30] Y. Wu and K. He. Group normalization. In *The European Conference on Computer Vision (ECCV)*, September 2018.