Learning to Reconstruct Symmetric Shapes using Planar Parameterization of 3D Surface

Hardik Jain, Manuel Wöllhaf, and Olaf Hellwich
h.jain@campus.tu-berlin.de, {woellhaf, olaf.hellwich}@tu-berlin.de
Computer Vision & Remote Sensing
Technische Universität Berlin, Berlin, Germany

Abstract

Shape priors have been a game changer to achieve robust 3D reconstruction. Prior knowledge encoded in trained networks has proven to be effective in generating images. Based on a similar paradigm, various methods were proposed to generate 3D shape from images. To generate a voxel or point cloud representation of 3D shapes these methods required adding an extra dimension to the deep network, to handle 3D data. Unlike these methods, we try to reconstruct 3D shape from images by using a parameterized representation of the shape. For a 3D model, the information is mainly concentrated on the surface. We perform iterative parameterization of the surface to obtain a planar representation. This representation is encoded with surface information to generate 2D geometry images, which can be conveniently learned using traditional deep neural networks without additional overhead. We propose an efficient iterative planar parameterization to represent regions of high Gaussian curvature in geometry images. Our experiments demonstrate that the proposed network learns detailed features and is able to reconstruct geometrically accurate shapes from single image. Our code is available at https://github.com/hrdkjain/LearningSymmetricShapes.

1. Introduction

With the advent of deep convolutional neural network in computer vision performing various image-based tasks, researchers have been exploring its application in 3D domain. In order to handle 3D models in form of voxels or point cloud, simple deep networks were extended with an extra dimension. The additional dimension enabled to solve various 3D problems in classification, reconstruction and segmentation. However, the increased computational overhead, restricted the resolution of 3D models that could be handled by such deep learning architectures.

In this work, we try to learn 3D surfaces for the task of 3D object shape reconstruction, taking into account that 3D objects are often perceived as a surface in 3D, e.g. of a genus-zero shape, circumferencing an empty or unknown while irrelevant volume. Representing 3D shapes in the form suitable for deep convolutional network is a difficult task. Because of the non-Euclidean nature of geometric objects, the handling of 3D surface meshes is not evident as a 2D image. We reduce the 3D problem to a 2D vision task by producing a regularized representation of 3D shape. First, geometry-image-based representation of object shape is generated and then a deep neural network is trained to learn this representation from object image.

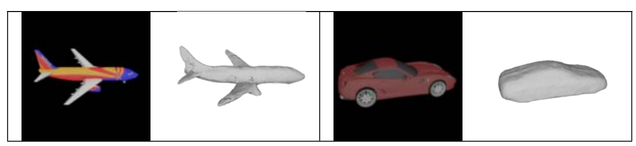


Figure 1. 3D reconstruction from single RGB image using our geometry image based deep neural network.

For this, our direct planar parameterization approach requires a manifold mesh in topology of disk. However, large 3D shape databases usually consist of non-manifold models, such as those handled by CAD databases. We adopt a heuristic approach to convert the non-manifold model to manifold mesh, preserving edges and sharp features of the model. To ensure consistent learning, the representations of different shapes need to be in correspondence. We achieve this by slicing the symmetric manifold surface at the plane of symmetry to obtain a consistent border for all heterogeneous shapes of same category.[1] Slicing the mesh into symmetric halves removes redundant information and provides a mesh in topology of a disk. The sliced mesh is then parameterized along its border in two stages. Initially, the 3D border is mapped to square planar border and then the inner vertices are parameterized. We adopt an iterative ap-

[1]Surprisingly many technical as well as natural objects have at least one symmetry plane.

proach while minimizing area distortion in the planar representation. The parameterized mesh is then encoded with 3D vertex location on a uniform grid of the geometry image. This encoded image is regarded as the 2D representation of the 3D object. Then, a deep residual network with encoder-decoder structure is trained on a larger number of object images in order to minimize the loss w.r.t. the geometry image.

Rest of the paper is arranged as follows, section 2 discusses the work related to 3D reconstruction and mentions how our method is different than the existing works. In section 3, we discuss the procedure involved in our method, including the iterative parameterization and deep network architecture. Results of various experiments are mentioned in section 4. We present our conclusions in section 5.

2. Related Work and Our Approach

Reconstructing 3D shape from images goes back to very beginning of the computer vision. Traditional methods try to reconstruct 3D surface from multiple images by establishing feature correspondence and solving epipolar geometry. Learning based methods gained attention in the last decade, where the task of 3D reconstruction was dealt as supervised problem of mapping images to 2.5D maps [23]. With the advent of deep learning, prior knowledge in the form of trained networks became popular. Ability of convolutional neural network to perform tasks like classification and generating images, motivated the extension of these techniques to 3D.

Modern approaches exploit the deep network for 3D shapes by adding an extra dimension to the network. Some of the pioneer works include 3D ShapeNets [27], VoxNet [17], which try to use deep network architecture for 3D shape recognition and completion. OctNet [21] enabled higher resolution handling of data by proposing an octree based network, they utilized this model for task of 3D object classification, orientation estimation and point cloud labeling. 3D recurrent neural network was utilized by Choy *et al*. [3] to generate 3D shapes from single or multiple images. Girdhar *et al*. [8] learned a vector representation for 3D objects from images using TL embedding network. This learned representation was used to generate voxel from input CAD object image. However addition of an extra dimension in the deep network, restricted the resolution of the voxel occupancy grid that could be processed. Similar work involves using a deep hierarchical feature learning on point sets [20]. Despite the improved performance such representation fails to capture fine-grained geometry.

Unlike the other 3D network based methods, Wang *et al*. [26] used a graph-based CNN and progressively deformed an ellipsoid to obtain 3D shape from image. Pontes *et al*. [18] used graph embedding, to reconstruct the 3D model from single image. AtlasNet [10] learned to generate surface of 3D shapes represented as collection of parametric surface elements. Other single image based shape reconstruction methods includes, [15] which used a category specific deformable model. Their method used 2D keypoint annotations, without requiring the ground-truth 3D. 2D perception was also used in Mesh R-CNN [9], to infer 3D shape. Where, firstly a coarse voxel is estimated, followed by mesh refining using a sequence of graph convolution layers.

Similar to our approach, Sinha *et al*. [25] tried to learn the 3D surface from geometry images. Their method required generating spherical parameterization on genus zero surfaces, which was then cut to obtain a planar representation. To enable consistent learning of geometry images, they performed parameterization of single shape in a class and established correspondence of all other shapes to this base shape. However this approach tend to smoothen the sharp features of object, there by giving good base shape but without having sufficient representation of areas of large Gaussian curvature.

We try to solve the two problems of consistent parameterization and correspondence of geometry images differently in our work. First problem of consistent parameterization is resolved by slicing the mesh and the second problem is addressed by using weighted mask for learning geometry images (section 4). In this method, we circumvent the problem of correspondence by utilizing the underlying property of symmetric objects by slicing the object mesh along the most symmetric plane[2] thereby removing redundant information and obtaining a consistent border for all objects of the same category. Then, after slicing, the border becomes a key to correspondence between different meshes of same object class.

Planar parameterization of surface mesh inevitably introduces distortion [13, 24], various methods try to reduce area [4], angular [5, 7] or length [22] distortion in the parameterized representation. In this work, the main motivation for parameterization is to obtain a trainable representation of 3D shapes, so we restricted the border of the parameterized surface to a square. Fixed border parameterization have linear complexity and generally employ discrete solvers. However, discrete methods fail to reduce the parameterized distortion of surface mesh with large Gaussian curvature. Spherical parameterization [19] is another approach to address parameterization. However this method requires identifying a seam along which the sphere is cut to obtain a square border parameterization. [28] proposed a square border stretch-minimizing parameterization. They improve the parameterization gradually by minimizing weighted quadratic energy.

Our area-distortion minimizing approach is motivated

[2]The problem how to find the symmetry plane is assumed to be solved in this investigation.

by [28], where we iteratively try to reduce the overall area distortion by spreading the stretch iteratively. Distortion minimized square border parameterized surface is encoded with surface information to generate geometry image [11]. It, being in 2D allows to apply standard neural network architectures to learn shape representation of an object image.

3. Procedure

To reiterate, our deep network relies on the regular grid 2D representation (akin to image) of the 3D surface mesh. In this section, we discuss the heuristic approach of generating manifold surface, followed by iterative parameterization, and finally the deep architecture used in our method.

3.1. Dataset Creation

For a surface mesh *M*, represented by *V*, *F*, *E* which are set of vertex coordinates, faces, and edges respectively. Genus *m* of the surface mesh is given by Euler characteristics:

$$2 - 2m = |V| - |E| + |F| \tag{1}$$

where $|x|$ denotes the cardinality of x. Meshes not following Equation 1 are often referred to as non-manifold. These meshes have non-manifoldness in at least one of the vertices or edges. Almost all the models of large 3D shape datasets like ShapeNet [2] and ModelNet [27] don't follow Euler characteristics.

Our method requires a manifold mesh as the input, but there exists no direct conversion from CAD model to manifold surface mesh. As this conversion is not the central topic of the work, we adopt a two stage heuristic pre-processing for non-manifold CAD models. First, a dense voxel point cloud is obtained for the 3D model. High density voxelization, helps retain detailed features of the model. Poisson surface is reconstructed from these voxel point cloud, where the vertex normal's information is taken from the 3D alpha shape [6]. This pre-processing is essential to generate input suitable for the planar parameterization and is applied on all the 3D models. For the sake of simplicity we removed all non-zero genus models.

3.2. Area Distortion Minimizing Mesh Parameterization

The objective of mesh parameterization in this work is to obtain a consistent parameterized representation for different meshes of same class. Performing parameterization requires identifying a seam on the mesh surface. For any parameterization method, a mesh can have different parameterizations based on the choice of seam. For examples we refer the readers to supplementary material. To establish consistency in parameterization, correspondence between seams of different meshes needs to be established. However establishing this correspondence could be difficult for heterogeneous shapes of a class. Instead of relying on the dense point-to-point correspondence to mark seam, we obtain consistent seam on border by slicing the mesh along the most symmetric plane.

For objects like airplane, car, chair, tables etc, this plane divides the mesh in two equal halves. Along with an obvious border, slicing also removes the redundant information which can be easily obtained from duplicating the sliced mesh. Slicing modifies the topology of the model to the topological disk, suitable for direct planar parameterization. With the border of mesh identified, we now discuss the parameterization of the sliced mesh along its border.

Aim of mesh parameterization is to obtain a bijective mapping between triangular mesh $S_{\mathcal{T}} \in \mathbb{R}^3$ which is of disk topology, and a planar triangulation $\Omega \in \mathbb{R}^2$. Let us denote the vertices in $S_{\mathcal{T}}$ by $\boldsymbol{x} = (x, y, z)^{\mathsf{T}}$ and points in Ω by $\boldsymbol{u} = (u, v)^{\mathsf{T}}$. $\mathcal{V}_B$ and $\mathcal{V}_I$ denotes the set of b boundary vertices and n inner vertices of $S_{\mathcal{T}}$, corresponding points in Ω is denoted by $\mathcal{U}_B$ and $\mathcal{U}_I$ respectively.

Direct planar parameterization can be divided into two stages. The first stage performs boundary parameterization. The set of border vertices $\mathcal{V}_B$ is mapped to the square planar boundary $\mathcal{U}_B$ based on edge length in $S_{\mathcal{T}}$. With the boundary parameterization fixed, the positions of inner points $\mathcal{U}_I$ are obtained by solving Equation 2 for u and v separately. For detailed explanation we refer the readers to the surveys [13, 24] and supplementary material.

$$M\mathcal{U}_I = 0 \tag{2}$$

where $M = (m_{ij})_{i,j=1,\dots,n}$ is the $n \times n$ matrix defined for inner vertices of $S_{\mathcal{T}}$. Elements of M are given by

$$m_{ij} = \begin{cases} -w_{ij} & \text{if } j \in N_i, \\ \sum_{k \in N_i} m_{ik} & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}$$

where w_{ij} are referred to as the weight of the edge joining vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and N_i denote all the neighboring vertices of $\boldsymbol{x}_i$. The most widely known cotangent weights [5] are used for initialization of matrix M.

As the distortion in planar parameterization is inevitable, we try to iteratively reduce it by minimizing overall area distortion of the parameterized mesh. Area distortion is calculated from Equation 3, where $\mathcal{T}$ and τ denote all triangular faces of $S_{\mathcal{T}}$ and Ω, respectively. $A(T_i)$ and $a(t_i)$ denote the area of i^{th} triangle in $S_{\mathcal{T}}$ and its corresponding triangle in Ω.

$$E_A(\mathcal{U}) = \sum_{T_i \in \mathcal{T},\, t_i \in \tau} \left| \frac{A(T_i)}{\sum\limits_{T_i \in \mathcal{T}} A(T_i)} - \frac{a(t_i)}{\sum\limits_{t_i \in \tau} a(t_i)} \right| \tag{3}$$

To reduce the parametric distortion in regions of high curvature, we redistribute the local L_2 stretch [22]. Weights of

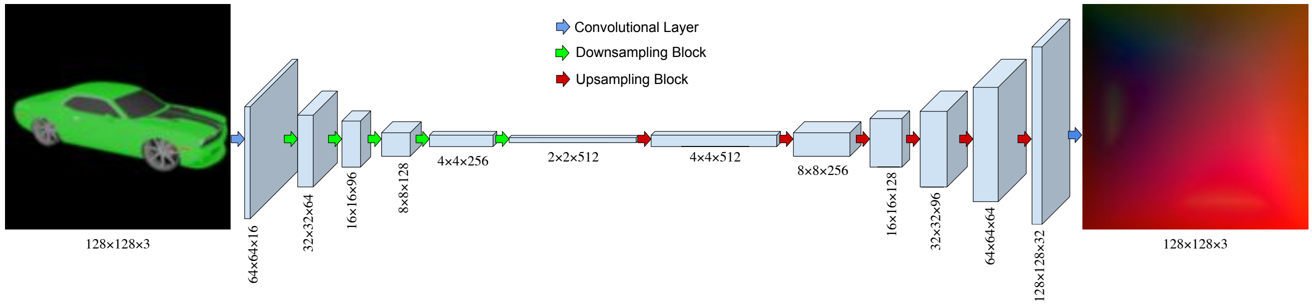


Figure 2. Deep neural network structure used for learning geometry image from the RGB image.

the $(k+1)^{th}$ iteration w_{ij}^{k+1} are modified according to the Equation 4 by the edge stretch σ_{ij}^{k} of the k^{th} iteration.

$$w_{ij}^{k+1} = w_{ij}^{k} / \sigma_{ij}^{k} \tag{4}$$

Equation 2 is then re-solved for the weights of $(k+1)^{th}$ iteration to obtain new parametric positions $\mathcal{U}_I^{k+1}$ for inner points. We iteratively improve weights, until the area distortion over the parameterized surfaces mesh is minimum. Please check supplementary material for pseudo code of the algorithm.

Obtained planar parameterization is an area distortion minimized representation of surface mesh. However the planar mesh is still irregular, so it is projected to a uniform grid and encoded with vertex locations to obtain geometry image [11]. Vertex position encoded geometry image holds the geometry of 3D mesh in a 2D representation.

3.3. Deep Network Architecture

In a vertex encoded geometry image representation, edge information is inherent. A pixel in geometry image corresponds to a vertex in the 3D shape, the neighbouring vertices of which are the four adjacent pixels in the geometry image. Learning the correct geometry image pixel value would automatically provide the edge information from the neighbouring pixels. So we focus our network to learn the pixel value correctly without the need to learn the edge connectivity explicitly. This reduces the problem to learning an image from other image using a simple mean square loss applied on the geometry image.

We used an auto-encoder like network structure motivated by [25], to learn geometry images from RGB images (see Figure 2). As in [25] we used residual down-sampling and up-sampling blocks [12] using 62 convolutional layers (plus 22 1×1 convolutional layers for projection) framed by one input and one output convolutional layer. Each down-sampling and up-sampling block contains three residual blocks. The first residual block of each applying the down/up-sampling using projection connections. We trained the model using Adam optimizer [16]. Our source code based on TensorFlow [1] is available online and includes all architectural details and parameter choices.

To enable learning of sharp geometry features, we modified the network loss function to accommodate weighted curvature mask C. Equation 5 shows the modified loss function, where f is the network response to the input RGB image $I^i(\theta)$, with sample number i and viewpoint denoted by θ. G_V^i denotes the corresponding vertex encoded geometry image for sample i.

$$min \sum_{(i,\theta)} ||C^i.(f(I^i(\theta)) - G_V^i)||_2^2 \tag{5}$$

Weighted curvature mask C is calculated from vertex normal encoded geometry image, G_N. Pixels of G_N contains vertex normal information, variation in angles of these normals in local neighborhood gives the curvature information. The single channel weighted mask C contains the average angle computed from the 8-connected pixels of G_N. Such a mask, captures region of large Gaussian curvature thereby enforcing the network to learn these regions.

We obtain a single geometry image and curvature mask for a 3D shape and train it against all the different views of the object shape. Therefore the G_V^i and C^i has no influence of viewpoint. This way, we are able to learn viewpoint independent geometry of the shape from image, which is required for generalization.

4. Experiment

For evaluation of our method, we used airplane and car models from ShapeNet [2] database and compare against geometry images based method of [25] and graph CNN based method of [26]. We initially selected 1930 airplane models and 2160 car models, however for fair comparison we restricted the size of our database to the models processed by [25]. We randomly separated these models into train-validation-test splits of 80-10-10%. To generate RGB images, the 3D models were rendered from different viewpoints with black background. Using Mitsuba renderer [14] each model was rendered for four elevation angles (0°, 15°, 30°, 45°) and 24 azimuth angles equally spaced in 360°.

4.1. Parameterization Analysis

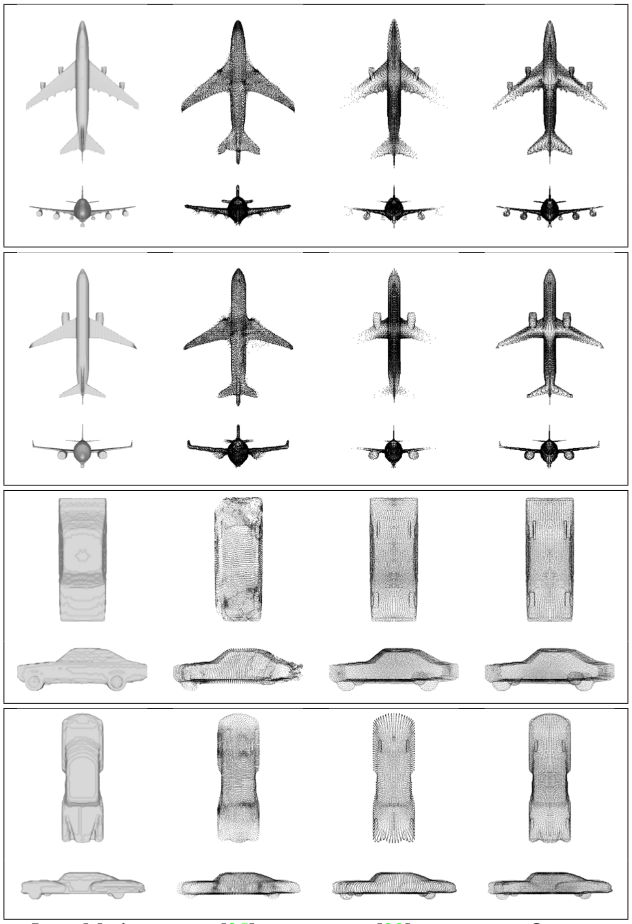


Figure 3. Comparison of different parameterization techniques based on output point cloud (obtained from geometry image), shown from two different viewpoints.

As mentioned previously our iterative parameterization scheme tries to minimize area distortion, thereby 3D mesh triangles are not significantly distorted in 2D parameterized representation. To compare different parameterization methods, the input shape is parameterized and encoded with vertices from 3D shape to obtain geometry image. Pixel of this image holds vertex location information and the same is shown in Figure 3 for different parameterization methods. Output is shown as point cloud to illustrate the distribution of vertices in re-meshed shape. Spherical parameterization of [25] distributes the vertices more uniformly however lacks to capture the sharp features. Stretch minimization approach [28], stuck in local minima and fails to parameterize the corners of the shape. Unlike the other two approaches, our proposed parameterization method tries to distribute the vertices more uniformly while covering the regions of large Gaussian curvature.

Method	RMS distance	
	Airplane	Car
Spherical parameterization [25]	0.011 ± 0.008	0.009 ± 0.004
Stretch minimization [28]	0.043 ± 0.021	$\mathbf{0.004 \pm 0.002}$
Proposed approach	$\mathbf{0.005 \pm 0.006}$	$\mathbf{0.004 \pm 0.002}$

Table 1. Average RMS distance measured from input mesh to point cloud obtained from geometry image.

Metric comparison between the three parameterization methods is also performed (Table 1) by computing RMS distance between the input ground truth mesh and the re-mesh obtained from geometry image. Proposed parameterization outperforms other methods specially for object class airplane which has more curvature compared to car.

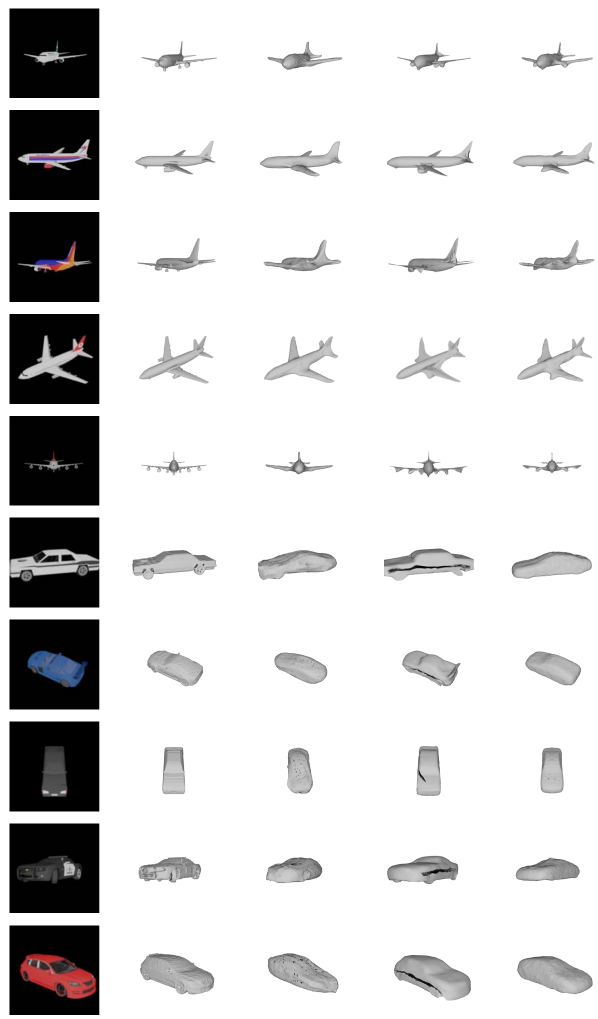


Figure 4. 3D shape reconstruction from RGB image on test data

4.2. 3D Surface Reconstruction from Single Image

We train our networks with all the 96 views of training dataset and perform evaluations on test models not used in training. For comparison we also train our network with geometry images (GI) of [25] using mask proposed in their approach. Output of our deep network is geometry image, pixel value of which holds the 3D shape of the model. Point cloud output of the network is meshed for visualization. Comparison is also performed with recent single image based approach of [26]. For fair comparison, all the models are trained with same data for 50 epochs. To avoid over-fitting early stopping was used.

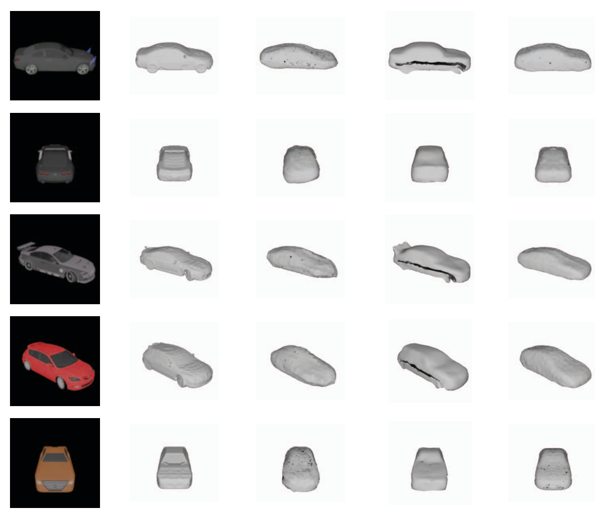
First evaluation is performed on images of test models rendered from the viewpoints which are used in training. Figure 4 shows the response of these networks. In this test scenario, our approach as well as Pixel2Mesh approach are able to perform detailed reconstruction. Unlike [25] and our approach, which uses surface representation to learn shapes, Pixel2Mesh uses sampled point cloud as the ground truth reference. In case of cars, this point cloud contains inner geometry information (seats) which is not visible in the image, but is used for learning shapes. Pixel2Mesh approach tries to deform an ellipse while maintaining the genus zero surface, which we believe is the reason behind high Gaussian surface bend on one side of the reconstructed surface. The output mesh is still genus zero however it also tries to incorporate the inner (not visible) geometry of the shape. We couldn't find any pre/post-processing applied in their original method to overcome this problem. Similar response could be observed in airplanes near the connection between wings and body.

To evaluate the generalization ability of the trained networks, we tested the networks on images of test models rendered from 24 additional view points not used in training. These viewpoints were equally spaced in azimuth at an elevation angle of 22.5°. Figure 5 and 6 shows the reconstruction on some example images. It can be observed that network trained on our geometry images are correctly able to retrieve geometric information like the size of the airplane body, tilt of the wings, bends on the car surface. Unlike GI of [25], our network is able to map sharp features from the input image like the tail and the engine of the model to the geometry image. We believe that the use of weighted curvature mask in training the network, helped the network learn regions of large Gaussian curvature. Geometry images of [25] generate an overall airplane shape but lacks to map detailed features from the input image. Pixel2Mesh approach holds generalization ability to an extend however fails in some airplane models (example 3 and 4 of Figure 5).

Quantitative evaluation of shape reconstruction was performed in terms of RMS distance between reconstructed point cloud and ground truth mesh. Average of this measure is shown in Table 2 for the three methods. On the trained viewpoints Pixel2Mesh outperforms our approach (in case of airplanes) but is not able to generalize for new viewpoints.

Input Image Ground Truth GI of [25] P2M [26] Ours

Figure 5. 3D shape reconstruction from RGB image rendered from untrained viewpoints for airplanes.



Input Image Ground Truth GI of [25] P2M [26] Ours

Figure 6. 3D shape reconstruction from RGB image rendered from untrained viewpoints for cars.

Our method is able to learn meaningful shape reconstruction from images. To illustrate this we perform shape reconstruction for images downloaded from internet (Figure 7). These images were cropped to match the input size. As the networks were trained on images with no background, we segmented out the background from these im-

Method	RMS Distance			
	Airplane		Car	
	Trained viewpoints	New viewpoints	Trained viewpoints	New viewpoints
GI of [25]	0.0382 ± 0.0205	0.0580 ± 0.0323	0.0361 ± 0.0347	0.0369 ± 0.0423
P2M [26]	$\mathbf{0.0112 \pm 0.0064}$	0.0412 ± 0.0118	0.0247 ± 0.0065	0.0323 ± 0.0088
Ours	0.0221 ± 0.0122	$\mathbf{0.0239 \pm 0.0108}$	$\mathbf{0.0185 \pm 0.0058}$	$\mathbf{0.0186 \pm 0.0059}$

Table 2. Average RMS distance from ground truth mesh to reconstruction performed on single image of test models.

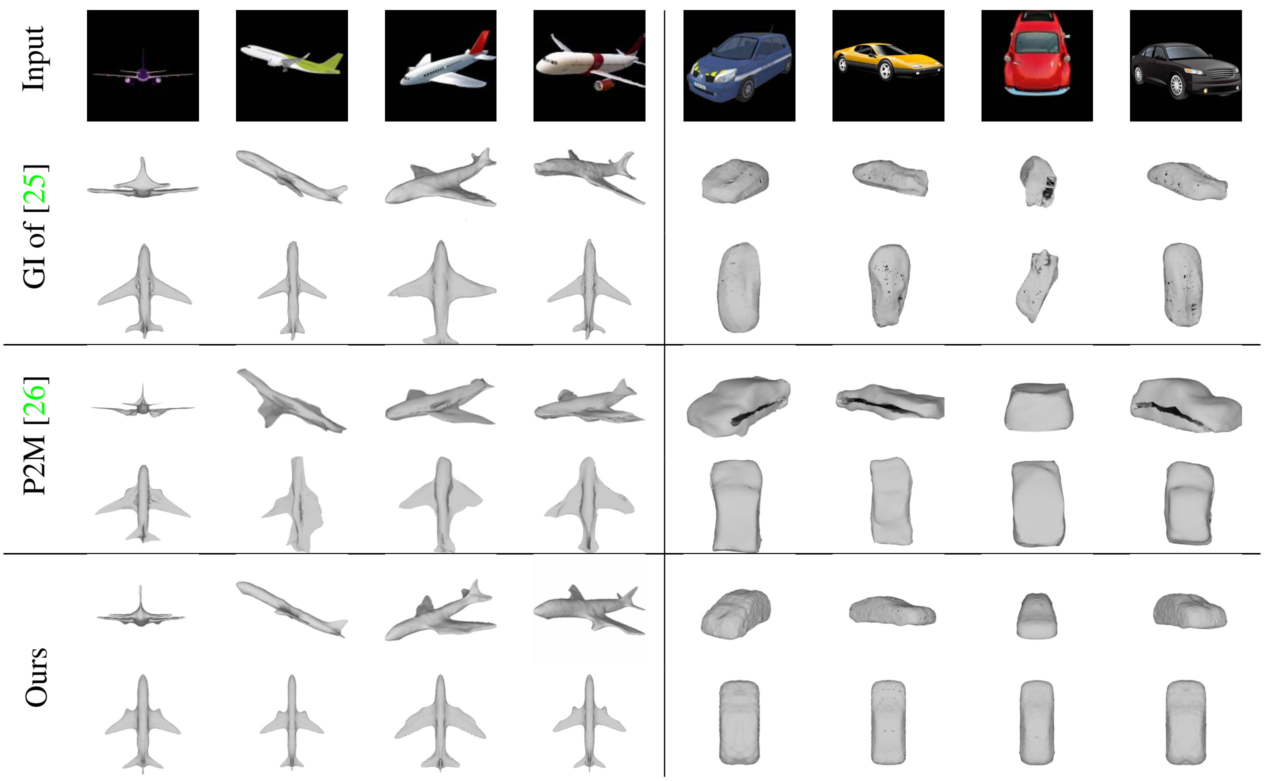


Figure 7. 3D shape reconstruction from images downloaded over the internet.

ages before feeding to the network. No additional fine-tuning was performed. GI of [25] generates the mean-shape without sharp features for airplane model and fails to reconstruct some cars correctly. Pixel2Mesh approach is not able to learn object specific information and generates shapes which look good enough only from viewpoint of the image. Same models when shown from alternate viewpoint, illustrates the non-symmetry in their reconstruction. Self occlusion in the RGB images restricts this approach from reconstructing the region not visible in the image. Using symmetry information in our approach helps retrieve shape information even though it is not completely visible in the input RGB images. Unlike the approach of [25] and [26] which uses view point specific ground truth for training the model, our method uses model specific ground truth geometry image. This way we believe that we avoid learning a viewpoint specific lookup table, which fails to deliver when shown images from viewpoints not used in training.

5. Conclusion

In this work, we try to reconstruct 3D shape surface from single image using deep residual network. Our method relies on the symmetry of an object to establish correspondence and obtain border for parameterization. The proposed network tries to learn meaningful representation from input images and preserves sharp features. Along with unseen images from trained viewpoints, we also tested our network for unseen viewpoints. Experiments conducted on random images collected from internet shows robustness of our method. In future works, efforts would be made to overcome the restriction of symmetry in the model by establishing surface correspondence. Another possible direction would be developing neural networks that could learn multiple shape categories simultaneously.

Acknowledgment

Hardik Jain would like to acknowledge DAAD (German Academic Exchange Service) for providing financial support for the studies.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 4

[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. *arXiv preprint arXiv:1512.03012*, (arXiv:1512.03012 [cs.GR]), 2015. 3, 4

[3] 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 *European conference on computer vision*, pages 628–644, 2016. 2

[4] M. Desbrun, P. Alliez, U. S. C. Inria, M. Meyer, and P. Alliez. Intrinsic Parameterizations of Surface Meshes. *Computer Graphics Forum*, 21(3):209–218, 2002. 2

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*, 6:173–182, 1995. 2, 3

[6] E. P. Edelsbrunner, Herbert and Mücke, H. Edelsbrunner, and E. P. Mücke. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994. 3

[7] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003. 2

[8] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a Predictable and Generative Vector Representation for Objects. In *European Conference on Computer Vision*, pages 484–499, 2016. 2

[9] G. Gkioxari, J. Malik, and J. Johnson. Mesh R-CNN. In *International Conference of Computer Vision*, 2019. 2

[10] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[11] X. Gu, S. J. Gortler, and H. Hoppe. Geometry Images. *ACM Transactions on Graphics*, 21(3):355–361, 2002. 3, 4

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[13] A. Hormann, Kai and Lévy, Bruno and Sheffer. Mesh Parameterization : Theory and Practice. Technical Report December, 2007. 2, 3

[14] W. Jakob. Mitsuba, 2014. 4

[15] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision*, pages 371–386, 2018. 2

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 4

[17] D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[18] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes. Image2Mesh: A Learning Framework for Single Image 3D Reconstruction. In *Asian Conference on Computer Vision*, 2018. 2

[19] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics*, 22(3):340, 2003. 2

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2

[21] G. Riegler, A. O. Ulusoy, and A. Geiger. OctNet: Learning Deep 3D Representations at High Resolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[22] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture Mapping Progressive Meshes. In *Computer graphics and interactive techniques*, pages 409–416, 2001. 2, 3

[23] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image Ashoutosh Saxena, Min Sun, and Andrew Ng. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 2

[24] A. Sheffer, E. Praun, and K. Rose. Mesh Parameterization Methods and their Applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, 2006. 2, 3

[25] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. SurfNet: Generating 3D Shape Surfaces using Deep Residual Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6040–6049, 2017. 2, 4, 5, 6, 7

[26] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. G. Jiang. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. *European Conference on Computer Vision*, pages 52–67, 2018. 2, 4, 5, 6, 7

[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 2, 3

[28] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. A Fast and Simple Stretch-Minimizing Mesh Parameterization. *Proceedings of the Shape Modeling International*, pages 200 – 208, 2004. 2, 3, 5