Residual MeshNet: Learning to Deform Meshes for Single-View 3D Reconstruction

Junyi Pan[1], Jun Li[2], Xiaoguang Han[3], Kui Jia*[1]

[1]School of Electronic and Information Engineering, South China University of Technology
[2]University of Technology Sydney
[3]Shenzhen Research Institute of Big Data, the Chinese University of Hong Kong (Shenzhen)

eejypan@mail.scut.edu.cn jun.li@uts.edu.au hanxiaoguang@cuhk.edu.cn kuijia@scut.edu.cn

Abstract

This work presents a novel architecture of deep neural networks to generate meshes approximating the surface of a 3D object from a single image. Compared to existing learning-based 3D reconstruction models, our architecture is characterized by (1) deep mesh deformation stacks with residual network design, where a simple mesh is transformed to approximate the target surface and undergoes multiple deformation steps to progressively refine the result and reduce the residuals, and (2) parallel paths per deformation step, which can exponentially enrich the generated meshes using deeper structure and more model parameters. We also propose novel regularization scheme that encourages the meshes to be both globally complementary to cover the target surface and locally consistent with each other. Empirical evaluation on benchmark datasets show advantage of the proposed architecture over existing methods.

1. Introduction

In this paper, we address the problem of inferring 3D geometric information of an object from a single image. The problem is both challenging and enlightening. Most geometric attributes, including 3D shape, of an object are inherently ambiguous given its 2D observation from a single perspective. Yet human can use prior knowledge to infer 3D information of an object from one picture, which motivates learning-based 3D reconstruction from a single 2D observation [28]. There are a few recent attempts in the direction of learning deep models to recover 3D surfaces of objects, where 3D surfaces are modeled/approximated using voxel [2, 19, 26], point cloud [5], or mesh [6, 27] based representations. Among them, point cloud based representation has avoided the quantization error induced by voxel based one, and mesh based representation further provides relations of local neighborhood, which enables piece-wise planar approximation of the continuous object surface. In this work, we are interested in learning to generate object surface meshes from a single image.

Figure 1. Given a single image as input, our method can generate the corresponding high resolution mesh output.

Formally speaking, the surface of an object approximates a 2D manifold embedded in the 3D Euclidean space. As an approximate modeling of the manifold, a polygon mesh is defined by vertices, edges between neighbored vertices, and also faces enclosed by connected edges. Except for the inverse challenge of inferring 3D information from 2D observations, it is a challenge of combinatorial optimization itself to construct meshes from a given set of vertices that are assumed to be sampled from the manifold. In fact, given an underlying manifold, there may exist multiple ways of meshing that approximate the manifold equally well. Given these challenges, we follow in this work the compromised problem setting [6] where in addition to the RGB image, a pre-defined mesh, usually defined in a 2D coordinate space, is also provided as input of the learning system. Consequently, learning to generate a mesh of an object surface amounts to learning a *parametrization* that transforms points/vertices in the input mesh onto the surface, where edges and faces of the input mesh are preserved.

The existence of multiple ways of meshing that approx-

*Corresponding author

2475-7888/18/$31.00 ©2018 IEEE
DOI 10.1109/3DV.2018.00087

imate an object surface equally well arguably implies that given a ground-truth mesh, one is not supposed to learn to generate a mesh that is identical to the ground-truth one; instead, one is expected to generate one or multiple meshes that can model the object surface as well as the ground-truth one does. Learning to generate multiple meshes has the following appealing advantages: (1) *geometrically*, these multiple meshes may serve as complementary piece-wise planar approximations to local tangent spaces of the manifold/surface, which as a collection can potentially contribute to a finer geometric approximation of the surface; and (2) *topologically*, multiple mesh parametrizations are necessary to generate an object surface of complex structures, e.g., those with non-disk topologies.

To achieve these advantages, we propose in this paper a novel, efficient, and effective deep model, termed Residual MeshNet (ResMeshNet), for generating a collection of *globally complementary* and *locally consistent* meshes from a single RGB image (see Figure 1). Figure 2 gives an illustration of the architecture. Our proposed ResMeshNet consists of stacked blocks of multi-layer perceptrons (MLPs). Each block of MLPs is a processing *stage*. The initial stage serves as an image encoder [9] that extracts *shape features*. Each successive processing stage consists of a single or multiple MLPs, each of which takes as input a mesh (coordinates of vertices combined with the shape features) and outputs its deformed version. The output of each stage is a set of meshes that collectively approximate the target surface. When $n > 1$ MLPs are used in a block/stage, our method essentially increases the resulting meshes by a factor of n. A shortcut connection is used between two successive blocks, similar to the way in [9]. Our use of shortcut connections has two benefits: in the forward pass, the shortcut connection adds (vertex coordinates of) a mesh in the current block to the output of each of the MLPs in the successive block, and consequently each MLP of the successive block only learns mesh residuals/offsets w.r.t. the one sent by the shortcut; in the backward pass, the shortcut connection sends training error signals directly to each of the intermediate blocks of ResMeshNet, resulting in geometrically consistent mesh generation across blocks. Each path of MLPs from the first stage to the last stage of ResMeshNet defines a parametrization that transforms vertices of an input mesh to the target surface, where the input mesh can usually be specified easily, e.g., a regular grid tessellating the 2D unit square as used in [6]. Compared with [6], our ResMeshNet is also much more efficient in terms of model complexity since the number of MLPs grows only logrithmically with that of the required output meshes.

To train our proposed ResMeshNet, we use Chamfer distance based objective [5, 6] that encourages the resulting meshes of an object surface *as a whole* to be consistent with vertices of the ground-truth one. Globally geometric complementarity of our method is thus achieved by the compound factors of residual learning and the above training objective. To achieve locally geometric consistency among resulting meshes, we additionally propose a regularizer that aims to make the resulting meshes be consistent with each other by penalizing the deviations of individual vertices of any mesh to their respective projections onto the closest faces of other meshes.

2. Literature Review

Deep neural networks (DNN) have enjoyed intensive research interest recently [12]. DNN-powered machine vision systems have achieved impressive success [11, 24, 25]. On the other hand, current machine vision systems cannot rival biological vision in 3D reconstruction from images, which is common and useful but challenging due to the information loss caused by the perspective projection during the imaging process. Traditional approach employs a pipeline that detects feature points in multiple images, followed by descriptor extraction and matching and reconstruction using multiple view perspective geometry [8]. Many widely used feature detectors and descriptors are hand-crafted according to heuristics such as affine transform invariance [20, 13]. These systems have worked for specific tasks and scenarios [16]. But there are practical challenges for which heuristic rules are less obvious and manual improvement or adjustment is difficult to design [29]. Building automatic geometry discovery system has made progress in certain operations: DNN-based models have been proposed to detect and encode features from images [14, 15]. Nevertheless, fully automatic 3D reconstruction from a single-view image remains an open problem [6, 2].

At the technical level, there still lacks a universally appropriate formulation for the 3D reconstruction problem that readily suits the end-to-end learning paradigm. Important design decisions include the representation of the 3D target, the evaluation criterion to adjust the model, and the overall network architecture to extract image information and predict the 3D target.

First, an important choice in designing the model is the specific format of the reconstruction target, 3D geometry. An entity in the 3D space has many attributes, and the suitable representation depends on which aspects are relevant to the task. Volumetric models are convenient to express space occupancy [2]. High-resolution 3D grid is expensive for computation and storage, for which schemes of generating voxels with adaptive sizes have been proposed [7, 26, 19]. A point cloud is another commonly used format to represent a 3D object, consisting of points sampled from the object surface [22]. A set of 3D points is flexible and able to express complex geometry, easy to produce and facilitates system implementation. However, the lack of intrinsic structure in the generated points can make subsequent applications dif-

ficult. For certain analytic tasks such as object classification, there are recent advances in designing DNN especially tailored to process point clouds [17, 18]. We have chosen to represent a 3D object by using meshes approximating the surface. Meshes are convenient for rendering and many other 3D modelling tasks, but less obvious to produce from an input image . An effective approach is [6], where a DNN is employed to deform a primitive mesh in the 2D space (regular grid of the unit 2D square) and produce a 3D embedding. The method is extensible to multiple meshes. In [27], one ellipsoid mesh has been deformed and upsampled progressively in a multi-step network, which employs generalized convolutional neural network [23] to aggregate features in neighboring vertices.

Training adjusts the network parameters to minimize the distance between the generated and the ground-truth 3D models and thus requires a distance metric between two 3D models. A number of useful criteria are defined in terms of two point sets sampled from the two 3D models, e.g. F-score using the notion of precision and recall [10], Chamfer distance [5] or Earth Mover's distance [21]. When the 3D models are of the mesh format, the surface quality can also be assessed [3]. We use the Chamfer distance to train our model, as the minimization in the computation of point-to-set distances globally couples all generated meshes and encourages complete coverage of the target surface. However, Chamfer distance does not account for intersections or gaps between multiple meshes, for which we proposed a specialized regularizer to improve reconstruction quality.

A characteristic design of the proposed architecture is that the incremental deformation induced by the shortcut connection between the earlier and later deformation stages. The idea of progressive modelling in a hierarchy has been proven effective [9]. For 3D meshes, [27] employed shortcut link across the graph (mesh) edges to connect the vertices. While we adopt shortcut to connect vertices belonging to the deformation results of different stages. Note [9] was proposed as a framework of image classification. We also adopt the network to encode an image as the shape features.

3. Residual MeshNet for Single View 3D Reconstruction

3.1. Problem Definition

Given an image of an object, the target of 3D reconstruction is the surface of the object. We employ mesh as a discretized practical representation of the object surface. A mesh is a specific type of graph with regular arrangement of the vertices as an array and edges between neighboring vertices. We consider the 3D reconstruction problem as formulating deformations of a *primitive mesh* $\mathbf{p}^{(0)}$ consisting of 2D vertices tessellated on $]0,1[^2$. Each deformation maps every vertex in $\mathbf{p}^{(0)}$ to a point in the 3D space, while maintaining the mesh edges between mapped vertices. In the discussion below, we will mostly refer a mesh to such a deformed version of $\mathbf{p}^{(0)}$ via adding offsets to the vertices. When there is no danger of ambiguity in the context, mesh may also represent the generic meaning of grid-like object 3D surface. The goal is to have the deformed meshes $\mathcal{M} = \cup_k \varphi_k(\mathbf{p}^{(0)})$ approximate the manifold $\mathcal{S}$, where φ_k is one deformation map. Formally, consider the recall error for a point on the target manifold and the precision error for a point on the deformed meshes respectively,

$$\epsilon_R(\boldsymbol{p}) := \inf_{\boldsymbol{q}\in\mathcal{M}} \{D(\boldsymbol{p},\boldsymbol{q})\}, \boldsymbol{p} \in \mathcal{S} \tag{1}$$

$$\epsilon_P(\boldsymbol{q}) := \inf_{\boldsymbol{p}\in\mathcal{S}} \{D(\boldsymbol{p},\boldsymbol{q})\}, \boldsymbol{q} \in \mathcal{M} \tag{2}$$

where $\epsilon_R(.)$ and $\epsilon_P(.)$ are the recall error and precision error respectively. In practice, a manifold is represented by finite number of points sampled from it, and the $\inf\{.\}$ take the minimal value from a finite set. The goal is to construct the deformation maps $\{\varphi_k\}$ to minimize the approximation errors, e.g. $\sup_p\{\epsilon_R(\boldsymbol{p})\}$ + $\sup_q\{\epsilon_P(\boldsymbol{q})\}$ or $\mathbf{E}[\epsilon_R(\boldsymbol{p})]$ + $\mathbf{E}[\epsilon_P(\boldsymbol{q})]$, where $\mathbf{E}[\cdot]$ is the expectation over a set using some measurement.

3.2. Residual MeshNet: Model

We use stacked deep neural network blocks, Residual MeshNet (ResMeshNet), to represent the deformation map from $]0,1[^2$ to 3D meshes approximating embedded surface. The deformation is modulated by a single image of the object. Figure 2 illustrates the overall architecture of a 3-stage reconstruction network. All information about the shape of the target object is encoded in a shape feature vector $\boldsymbol{x}$, which is extracted from the image by an encoder subnet. Given $\boldsymbol{x}$, the cascaded mesh deformation starts from the vertices in the primitive mesh $\mathbf{p}^{(0)}$. As a specific example of $\mathbf{p}^{(0)}$, we use a primitive mesh of 10×10 vertices evenly distributed on the 2D unit square. After multiple stages of parallel and incremental deformation, the network outputs multiple versions of deformed $\mathbf{p}^{(0)}$: $\varphi(\mathbf{p}^{(0)};\theta) \mapsto \{\mathbf{p}^{(L,1)}, \mathbf{p}^{(L,2)}, \ldots, \mathbf{p}^{(L,M)}\}$, where each $\mathbf{p}^{(L,\cdot)}$ is a set of 3D points corresponding to a particular deformation of $\mathbf{p}^{(0)}$ and M is the number of total deformations.

Our proposed ResMeshNet consists of stacked blocks of multi-layer perceptrons (MLPs). Each MLP is composed of 4 fully-connected layers. The detailed architecture of MLP is shown in Figure 3. The t-th deformation block receives the coordinates of 3D vertices (for t≥2) from the previous block $\mathbf{p}^{(t-1)} \in \mathbb{R}^{3\times N_{t-1}}$, where N_{t-1} represents the number of vertices generated by the (t-1)-th block. The points in $\mathbf{p}^{(t-1)}$ are concatenated with the shape feature $\boldsymbol{x}$ to form the input vectors $\boldsymbol{u}^{(t)}$ to the MLP(s) in the block: $\boldsymbol{u}_i^{(t)} = [\mathbf{p}_i^{(t-1)})^T, \boldsymbol{x}^T]^T$, where i is the index for the point

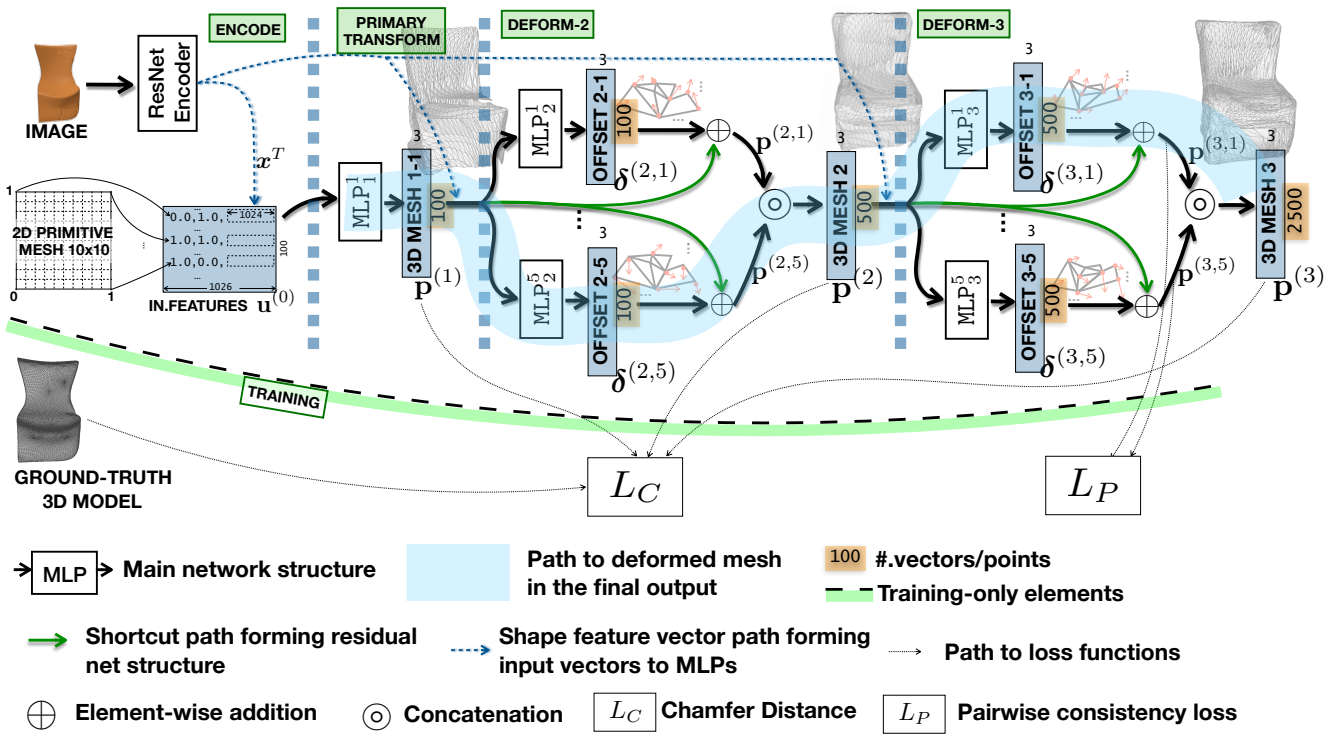

Figure 2. Diagram of ResMeshNet Architecture. The figure shows the structure, processing and training workflows of ResMeshNet. The legend includes the interpretation of some important concepts, such as the shortcut connection making a residual net structure, the multiple application of Chamfer distance loss L_C and the proposed pairwise consistency loss L_P. See text for details. This figure is best viewed in color on a computer screen.

and corresponding input vector. The Block-t can have one or several MLPs. The j-th MLP in this block transforms $\boldsymbol{u}^{(t)}$ and produces 3D offset vectors $\boldsymbol{\delta}^{(t,j)}$ for the input vertices. The offset $\boldsymbol{\delta}^{(t,j)} \in \mathbb{R}^{3\times N_{t-1}}$ is of the same size as the input vertices $\mathbf{p}^{(t-1)}$, representing the refinement by the j-th MLP in block t. The corresponding output vertices is $\mathbf{p}^{(t,j)} = \mathbf{p}^{(t-1)} + \boldsymbol{\delta}^{(t,j)}$. An individual MLP represents one deformation step on a particular mesh. Let the number of MLPs in the block be m_t. Then the total output of block t is the collection of all the deformed vertices sets $\mathbf{p}^{(t)} := [\mathbf{p}^{(t,1)}, \mathbf{p}^{(t,2)}, \ldots, \mathbf{p}^{(t,m_t)}]$. Note the total number of vertices will increase by a factor of m_t : $\mathbf{p}^{(t)} \in \mathbb{R}^{3\times N_t}$ and $N_t = N_{t-1} \cdot m_t$. In the case shown in Figure 2, we use 5 MLPs in both the second and third blocks, which increase the number of vertices by 5 at each block. So we can finally get the output with 2500 vertices. Note that there is point-wise correspondence in $\mathbf{p}^{(t,j)}$ and $\mathbf{p}^{(t-1)}$, and thus each $\mathbf{p}^{(t-1)}$ retains the mesh structure inherited from the primitive mesh $\mathbf{p}^{(0)}$ via the chain of transformations $\mathbf{p}^{(0)} \rightarrow \mathbf{p}^{(1)} \rightarrow \ldots \mathbf{p}^{(t)}$. When t = 1, the first deformation block performs the *primary deformation*, which maps $\mathbf{p}^{(0)}$ to the 3D space and makes the foundation of subsequent deformations. As above, the inputs to the MLP are composed point feature $\boldsymbol{u}^{(0)}$, where an individual $\boldsymbol{u}_i^{(0)}$ is the concatenation of 2D point $\mathbf{p}_i^{(0)}$ and shape feature vector $\boldsymbol{x}$.

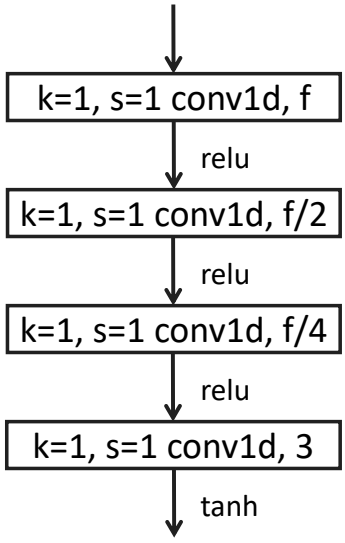

Figure 3. The Network Architecture of MLP. The number of input feature channels is denoted as f. k denotes the kernel size and s the stride.

For the shape feature vector, it incorporates the image information that is relevant to the 3D shape of the object. In theory, a generic image encoder extracting sufficiently informative $\boldsymbol{x}$ would do. In this study, we adopt a ResNet trained as an image classifier [9]. Though classification requires discriminative information, which can be different from the geometric information about the 3D shape in the image, the extracted image descriptor worked satisfactorily in our tests. Specialized encoder for 3D reconstruction from images may further improve the performance.

3.3. Training Objective

The training loss consists of two groups of terms,

$$L = L_C + L_P \tag{3}$$

where L_C is the Chamfer distance between the deformed vertices at *each stage*, $\mathbf{p}^{(t)}$, and a set of points sampled from the ground-truth surface, $\mathbf{q} \subset \mathcal{S}$,

$$L_C = \sum_{t=1}^{T} \Big(\sum_{x \in \mathbf{q}} \min_{y \in \mathbf{p}^{(t)}} \|x-y\|_2^2 + \sum_{y \in \mathbf{p}^{(t)}} \min_{x \in \mathbf{q}} \|x-y\|_2^2 \Big) \tag{4}$$

where T is the total number of stages. For each point, Chamfer distance finds the nearest point in the other point set, and sums the distances up [5]. Since our final output consists of multiple deformed meshes, we proposed the second group of loss terms, L_P, which is imposed on the final-stage deformation $\mathbf{p}^{(T)}$ to encourage the pair-wise consistency across the meshes

$$L_P(\mathbf{p}^{(T)}) = \sum_{p \in \mathbf{p}^{(T)}} D_P(p, \hat{\pi}_p) \cdot \mathbf{1}[\hat{p}_{\hat{\pi}_p} = p] \tag{5}$$

where $D_P(p, \pi)$ represents distance metric between a point $\mathbf{p}$ and plane π. The decorator $\hat{\cdot}$ has a special meaning of closest extraterrestrial object. By the term we mean, given a point p, $\hat{\pi}_p$ represents the closest face (a 2D plane) to p in the meshes $\mathbf{p}^{(T)} \backslash \{\mathbf{p}^{(T,j_p)}\}$, where $p \in \mathbf{p}^{(T,j_p)}$, i.e. $\hat{\pi}_p$ is the closest face to p from any mesh in $\mathbf{p}^{(T)}$ except the one to which p belongs. Symmetrically, for a face $\pi \in \mathbf{p}^{(T)}$, we can find its closest extraterrestrial point $\hat{p}_\pi$ within $\mathbf{p}^{(T)} \backslash \{\mathbf{p}^{(T,j_\pi)}\}$. The condition $\mathbf{1}[\hat{p}_{\hat{\pi}_p} = p]$ ensures L_P to only account for mutual closest point-plane pairs. I.e. the distance between p and $\hat{\pi}_p$ is counted only when p is also the closest extraterrestrial point for $\hat{\pi}_p$. Figure 4 illustrates how the regularization is calculated. The regularizer encourages meshes to be consistent to each other in overlapped areas.

3.4. Discussion on training and architecture

Loss and regularization for training. These loss terms have several interesting characteristics:

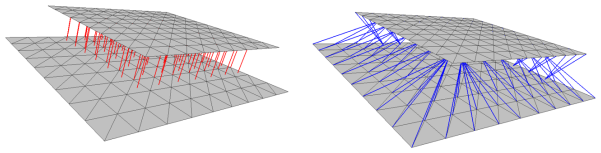

Figure 4. Illustration of the pairwise consistency regularizer L_P. This figure shows how the proposed pairwise consistency regularizer L_P in (5) works in a toy example. The red lines link the cross-mesh mutually nearest point-plane pairs, whose distances are penalized. The blue links show spurious pairs, where the mutuality condition in (5) does not hold.

1) Chamfer distance term L_C consists of $D_C(\mathbf{p}^{(T)}, \mathbf{q})$ as well as $D_C(\mathbf{p}^{(1,\dots,T-1)}, \mathbf{q})$. The later group of terms aims to improve the agreement between the intermediate deformed meshes to the ground-truth manifold. Those intermediate meshes are not part of the final output, thus Chamfer distances $D_C(\mathbf{p}^{(1,\dots,T-1)}, \mathbf{q})$ can be considered as training time regularization that encourages consistency that will affect all subsequent refinements.

2) Chamfer distance naturally corresponds to the mesh to manifold approximation error defined in (1) and (2).

3) The pair-wise consistency L_P is a regularization term defined independent of the ground-truth manifold, but encourages meshes to be consistent to each other in overlapped areas.

Architecture design. The ResMeshNet architecture has a number of distinctive attributes that facilitate shape estimation. In two successive deformation blocks, the later block employs any number of MLPs to make further deformation to the meshes produced by the the earlier block. This block-wise residual connection has interesting implications.

1) The approximation to the surface manifold is refined *quantitatively*: the number of meshes produced by the later block increases by a factor of m_t, the number of MLPs in the block, compared to the number of meshes given by the earlier block. Hence the resolution of the approximation is enhanced in terms of the collection of vertices and faces in the meshes. The total number of meshes (and vertices and faces therein) output by the architecture grows exponentially with the MLPs, given the MLPs are reasonably distributed among the blocks.

2) The approximation is also refined *qualitatively*: the later block produces offsets to add on top of the incoming meshes and aims to move the meshes closer to the target surface manifold.

A practical system can flexibly use a design that puts more weight on the two aspects of refinement. For example, if only one MLP is used in a block, the resolution of the input and output meshes will be the same and all refinement is done by the offsets on the vertices predicted by the MLP. The MLP can have a deep and sophisticated structure with

	A1	A11	A25	A125	wo/Shortcut	wo/Reg.	ResMeshNet
cabinet	3.11	3.19	3.06	3.10	3.19	3.09	**2.96**
table	3.36	3.23	3.20	3.23	3.27	3.15	**3.08**
chair	3.76	3.49	3.49	3.49	3.47	3.49	**3.19**
bench	2.60	2.35	2.51	2.45	2.30	2.53	**2.27**
cellphone	2.63	2.52	2.33	2.12	**2.10**	2.37	2.21
watercraft	2.84	2.62	2.68	2.58	**2.54**	2.74	2.59
monitor	4.37	4.38	4.87	3.90	4.55	4.34	**3.67**
car	2.92	2.79	2.73	2.61	2.59	2.68	**2.56**
couch	3.58	3.29	3.41	3.12	3.23	3.57	**3.07**
firearm	1.86	1.70	1.42	1.49	1.51	1.45	**1.39**
lamp	10.46	10.49	9.76	10.12	9.86	9.92	**9.46**
plane	1.79	1.65	1.64	1.58	1.62	1.71	**1.49**
speaker	6.16	6.52	6.26	**6.02**	6.06	6.49	6.65
MEAN	3.62	3.48	3.42	3.37	3.38	3.44	**3.23**

Table 1. Abalation study results. The table shows effects of various design characteristics of ResMeshNet: the shortcut connection, the every-stage CD loss and the hierarchical organization of MLPs. The Chamfer distance is in units of 10^{-3}.

many parameters to ensure necessary capacity to model the optimal adjustment on the existing meshes.

On the other hand, if a simple MLP structure is used. We can instantiated multiple such MLPs in a block. The hope is that though the capacity of individual MLP is limited, the multiple instances of meshes deformed by such MLPs can address different parts in the target manifold, and improve the overall approximation.

It is also noteworthy that the each mesh produced by the final block is the result of a chain of deformations. For example, the blue-shaded band indicates a path of deformation making the *parameterization map* [4] from the primitive 2D grid mesh to an output mesh approximating a part on the surface of target surface of a 3D shape. Along the path, each stage can be viewed as a distributed modelling of the shape manifold embedding, and the hierarchical composition of the stages helps capture rich geometrical structures using small-sized models, since the number of potential paths grow exponentially with the number of the stages. In the sense of modelling mesh deformations, the proposed architecture realizes the successful principle of deep learning, namely distributed and hierarchical representation. Our experiments verify that the proposed model achieves superb performance using fewer MLPs.

4. Experiments

Datasets We carry out our experiments on the widely used ShapeNet Core dataset [1]. The dataset includes shapes of 13 object categories, each of which has 1,000˜10,000 instances of an object 3D model and the corresponding rendered images. To make our results comparable with those of existing methods, we adopt the experiment setup used in [2, 6].

Architectural Details As shown in Figure 2, our ResMeshNet starts with an image encoder and has three stages of mesh deformation including the primary one. We use the same image encoder of ResNet-18 as in [6], whose final FC layer outputs a $1,024$-D shape feature vector. The three stages/blocks respectively contain $n_0 = 1$, $n_1 = 5$, and $n_2 = 5$ MLPs for mesh deformation. We also use the same MLP of 4 FC layers as in [6]. Thus we can finally get $1 \times 5 \times 5 = 25$ meshes transformed from the primitive 2D mesh. The primitive mesh is a 10×10 grid of vertices regularly sampled from the unit 2D square, and our final result of surface reconstruction contains $2,500$ 3D vertices during training. During inference, we always generate high resolution meshes with nearly $30,000$ vertices. We restrict our model to use 2,500 sample points from the ground-truth mesh during training to keep the optimization efficient. Compared to existing methods such as [6], our model complexity has been significantly reduced for a specified granularity of mesh vertices.

Towards Mesh based Evaluation We follow the literature and use Chamfer distance (CD) to approximately measure results of surface reconstruction. CD is defined between points from the ground-truth mesh and those from the resulting meshes. To be consistent with training, $2,500$ points from each ground-truth mesh are used for CD evaluation. In order to approach a mesh based evaluation, we also take a strategy of sampling more points from the ground-truth mesh and the resulting mesh, and compute the corresponding CD results. Such results provide better indication of mesh reconstruction quality.

Results of Controlled Studies

Shortcut Connection To examine the effect of shortcut connections between stages, we devise a model simply by re-

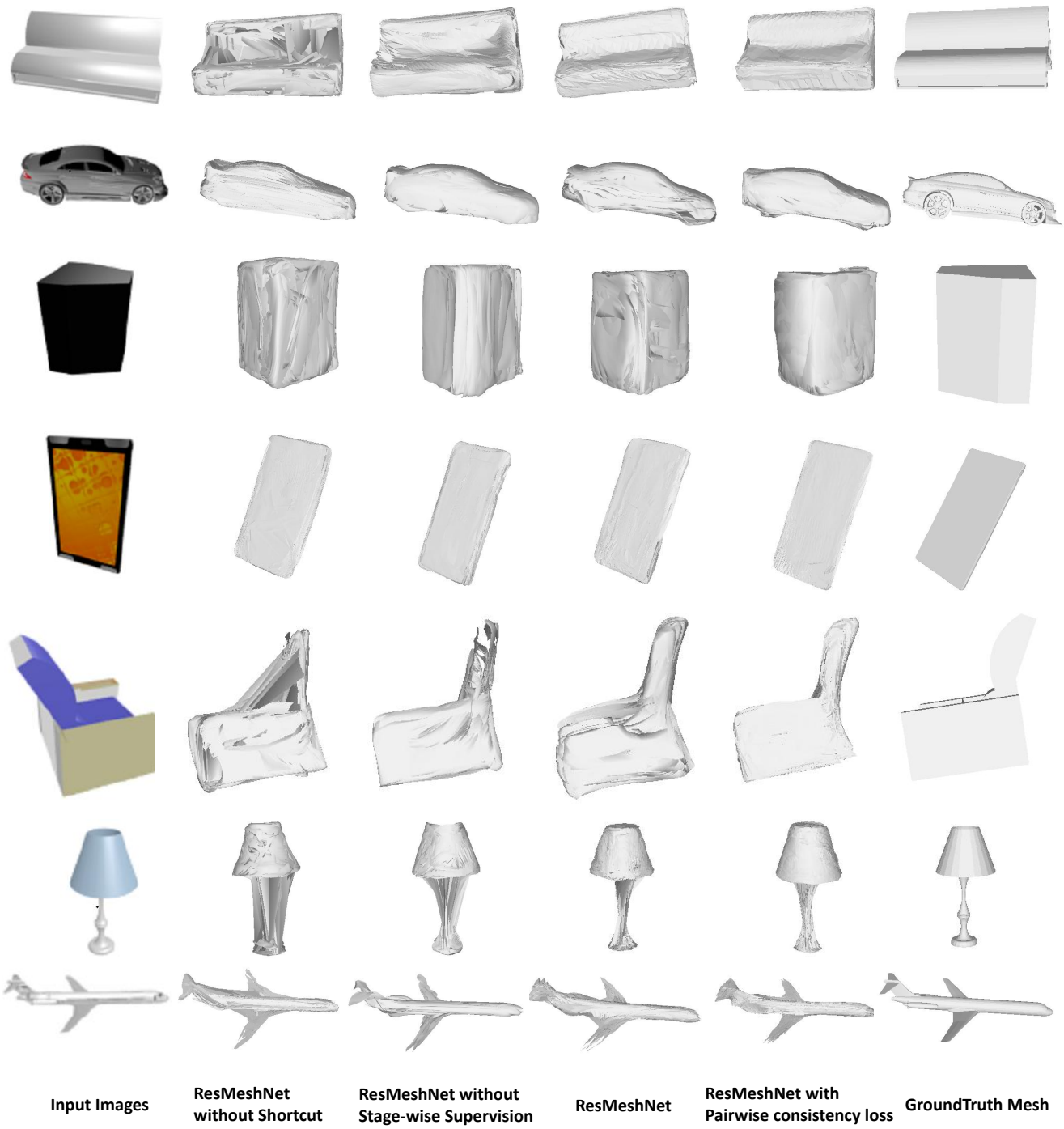

Figure 5. Visualization results of ResMeshNet and other controlled experiments.

moving the shortcuts in our ResMeshNet. Results in Table 1.Column-"wo/Shortcut" demonstrate the clear advantage of using shortcuts.

Stage-wise Supervision Our ResMeshNet produces at each stage an approximation to the target 3D surface, which enables stage-wise loss computation. We argue that stage-wise supervision may be beneficial to surface reconstruction. To investigate, we conduct experiments by training ResMeshNet with no use of supervision in the intermediate meshes. Results in Table 1.Column-"wo/Reg." corroborate our hypothesis.

MLP Hierarchy The hierarchical structure of ResMesh-

Net allows to exponentially increase the number of output meshes by linearly increasing that of MLPs. In Table 1.Column-A1, A11, A25 and A125, we quantitatively assess the performance gain of such a MLP hierarchy in terms of CD results against model complexities. While our ResMeshNet has the same complexity as that of AtlasNet11, our result is much better than theirs.

Mesh Evaluation More sampled points can represent finer details of a manifold. For test, in order to approach a mesh based evaluation , we sample more points from the ground-truth mesh and the resulting mesh to compare ResMeshNet and AtlasNet. The results sampling various numbers of points are shown in Figure 6. The CD-to-sample-number curves show that ResMeshNet has consistent advantage, and when more points are sampled, our method enjoys even more advantage over AtlasNet.

Comparison with Existing Methods We compare our results with existing methods of PSG and AtlasNet in Table 2. Our results are better than those of these methods. Table 2 also shows that with use of the regularizer (5), the results degrade slightly. However, the regularizer produces visually smoother results of mesh reconstruction as shown in Figure 5.

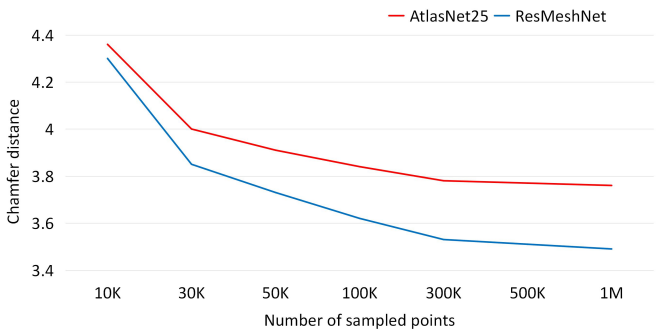

Figure 6. Results using Different Mesh Evaluation Settings. The results are on randomly selected 260 shapes from all 13 categories. Horizontal axis represents the number of 3D points sampled from ground-truth mesh and resulting mesh. The vertical axis represents the CD-loss in units of 10^{-3}.

	MEAN-CD
PSG	6.09
AtlasNet	3.42
ResMeshNet	3.23
ResMeshNet_PR	3.30

Table 2. Mean CD on the ShapeNet Core dataset. ResMeshNet_PR means ResMeshNet with pairwise consistency loss. The Chamfer distance is in units of 10^{-3}.

5. Conclusion

We have proposed a new deep neural network architecture to generate meshes representing a 3D object, for which only a single-view observation is available, by progressively transforming and deforming pre-defined simple meshes to approximate the object surface. The network consists of multiple deformation blocks stacked with cross-block shortcut connections, so the later deformation blocks model the residuals left by the earlier blocks. Each deformation block can employ a flexible number of MLPs and potentially increase the number of resulting meshes (by a factor of the number of MLPs). We also propose a regularization scheme that encourage the meshes to cooperate globally to approximate the target surface and to be consistent locally. Extensive empirical study has shown the effectiveness of the proposed method. Future research can be on explicit coordination between meshes to achieve better global parameterization of a surface manifold.

Acknowledgments

This work is supported in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No.: 2017ZT07X183), the National Natural Science Foundation of China (Grant No.: 61771201) and Shenzhen Fundamental Research Fund (Grant No.: KQTD2015033114415450).

References

[1] 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. 6

[2] 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 *ECCV*, 2016. 1, 2, 6

[3] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Comput. Graph. Forum*, 17(2):167–174, 1998. 3

[4] M. do Carmo. *Riemannian Geometry*. Mathematics (Boston, Mass.). Birkhäuser, 1992. 6

[5] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017. 1, 2, 3, 5

[6] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *CVPR 2018*, 2018. 1, 2, 3, 6

[7] C. Häne, S. Tulsiani, and J. Malik. Hierarchical Surface Prediction for 3D Object Reconstruction. 2017. 2

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 3, 5

[10] A. Knapitsch, J. Park, Q. Zhou, and V. Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017. 3

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012. 2

[12] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015. 2

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[14] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, pages 4829–4840, 2017. 2

[15] D. Mishkin, F. Radenovic, and J. Matas. Learning discriminative affine regions via discriminability. *CoRR*, abs/1711.06704, 2017. 2

[16] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, pages 9–16, 2009. 2

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, pages 601–610, 2017. 3

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. 2017. 3

[19] G. Riegler, A. O. Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 6620–6629, 2017. 1, 2

[20] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, pages 2564–2571, 2011. 2

[21] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. 3

[22] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, pages 3212–3217, 2009. 2

[23] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 3

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2

[25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. pages 4278–4284, 2017. 2

[26] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. In *ICCV*, pages 2107–2115, 2017. 1, 2

[27] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. 2018. 1, 3

[28] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *CVPR*, 2016. 1

[29] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: learned invariant feature transform. In *ECCV*, pages 467–483, 2016. 2