FuseNet:Fusing surface and volumetric representations for 3D Shape Synthesis and Analysis

Anonymous ICCV submission

Paper ID 2714

Abstract

Surface and volumetric representations of objects have been explored separately for 3D synthesis, however by combining these modalities we achieve more accurate results. Understanding the surface and volumetric representations jointly make shape representations more descriptive and informative. We propose a system of neural networks named FuseNet which is a conditional generative model that synthesizes objects from a specific class without reference to images or CAD models. One aspect of FuseNet is utilizing both surface and volumetric representations for shape synthesis and analysis. Meanwhile, objects are combinations of functional parts and part geometry is important for understanding shape space. In FuseNet, we introduce a part identifier module which learns part geometry to preserve part properties of 3D objects. To demonstrate the capability of learned shape representations from FuseNet, we performed shape retrieval and reconstruction. FuseNet outperforms state-of-the-art methods in shape synthesis and reconstruction. In shape retrieval, it is comparable with state-of-the-art methods.

1. Introduction

The technical advancement of 3D printing, virtual reality and augmented reality has brought significant interest in modeling and understanding 3D shapes to the vision and graphics communities. This increased interest has lead to the development of shape synthesis [2, 53, 18], 3D reconstruction [11, 8], and view synthesis [43, 35]. The emergence of neural networks and the creation of large-scale 3D shape datasets [6, 51] inspired researchers to conduct 3D representation learning and geometric analysis by using view-based projections [21], polygon meshes [41, 55, 31], point clouds [1, 37], and voxelized 3D shape data [15, 7]. In this work, we focus on voxelized 3D data in combination with surface data for modeling conditional generative models and learning compact shape representations.

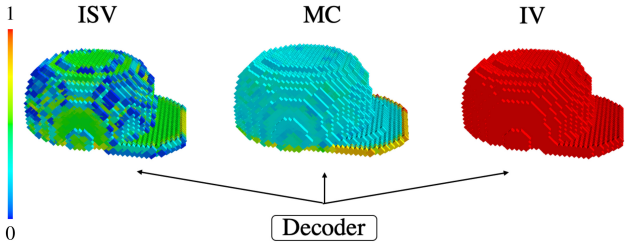

Figure 1: A decoder of FuseNet. FuseNet fuses surfaces and volumetric representations for learning compact shape representations. Through multi-task learning, the decoder predicts Intersected Surface area in a cubic Voxel (ISV), Mean Curvature (MC), and the Interior Volumetric representations (IV) of the object.

Earlier approaches treat the surface and volumetric representations of objects separately, however by combining these modalities we achieve more accurate results. Merging the curvatures and volumetric representations for 3D shapes has not been explored to the best of our knowledge. The surface and the volumetric representations of objects contain unique features and are complementary. The surfaces impose boundary information for a 3D object, enclosing a volume, and the volumetric objects determine interior geometry which is used for heat flow calculation [26, 33]. However, consolidating these two representations has not been well-examined for learning compact shape representations and synthesizing objects.

To show the effectiveness of fusing surface and volumetric representations of objects, we propose FuseNet, a conditional generative model that fuses the boundary information and volumetric representations of objects, for learning compact shape representations. FuseNet integrates surface and volumetric representations by encoding Sphere Bounded Mean Curvature (SBMC), Mean Curvature (MC), Intersected Surface area in a cubic Voxel (ISV), and Interior Volumetric representations (IV) of objects as an input and

decoding ISV, MC, and IV as an output. Sharing surface and volumetric information is crucial for learning a perceptual set of attributes, such as the connectivity of parts and details of local interior regions, hence reducing defects in synthesized objects as shown in Section 5.2.

In addition, part geometry is critical for understanding shape space, since objects are combinations of functional parts. However, learning part geometry is not a trivial task due to the high cost associated with labeling each part of the objects. Defining the boundary of each part is a complex problem since parts can be connected in many different ways. Current CAD datasets are not constructed to share parts information among similar models. This makes it even worse to distinguish between specific parts within an object. Additionally, creating a part segmented database requires much more effort than creating a classification dataset.

To alleviate the shortage of the dataset, we expand the dataset [20] by reshaping specific parts of objects as shown in Figure 2. Each part has their own unique shape and location for their purpose, which creates unique part geometry. For learning part geometry of objects, we propose a *part identifier* module to locate and recognize parts. With the understanding of part geometry, FuseNet enhances the fidelity of each part as evaluated in Section 5.2.

FuseNet is designed to perform multi-task learning to improve generalizability of the networks. The decoder module in FuseNet consists of a single decoder (see Figure 1) for synthesizing ISV, MC, and IV to share interior volumetric and boundary information of objects. In this way, the model learns surface properties to learn the interior volumetric representations of objects and vice-versa [47]. Additionally, the part identifier module estimates part geometry and shares the part information with other modules by backpropagating the meaningful gradients during training.

Our main contributions are summarized as follows:

1. We propose fusing surface representations and the volumetric representations of the 3D objects by consolidating the complementary aspects of SBMCs, MC, ISV, and IV, which are more informative for object synthesizing and object representation learning.

2. We introduce a single decoder for synthesizing ISV, MC, and IV of an object, which learns surface properties to learn interior volumetric representations, and vice-versa, during multi-task learning to improve the generalizability of FuseNet.

3. We propose $part\ identifier$ module, which improves the property preservation of each part of the synthesized objects. Learning part geometry is done explicitly by optimizing the part identifier module and expanding the dataset by adjusting functional parts of the objects.

4. We propose SBMCs to capture rapid local changes in mean curvatures. These are encoded into the voxel grids and used as an input modality, which compactly encodes the series of fast local changes of mean curvatures.

Toward this end, we demonstrate ablation studies and comparison experiments with other methods. We perform an object synthesis with a one-hot encoded class vector and a vector from normal distribution. For evaluating shape representations obtained from FuseNet, we performed two applications: shape reconstruction and shape retrieval. From our results, the proposed method outperforms state-of-the-art methods in conditional object synthesis and shape reconstruction and is also comparable with the state-of-the-art methods in shape retrieval.

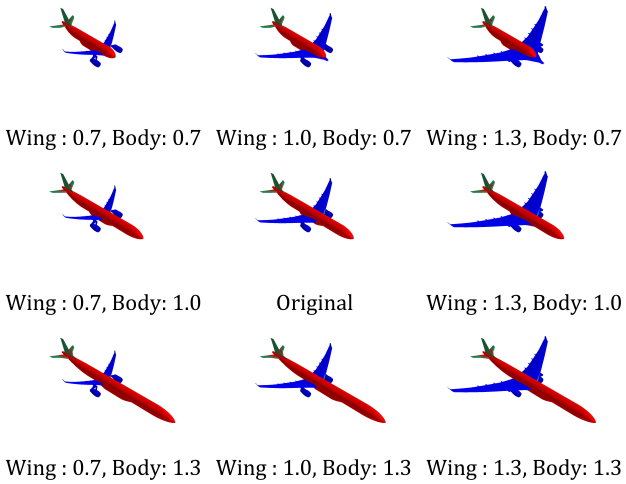

Figure 2: Mesh models are expanded by relocating vertices of parts to learn part geometry. Parts are numbered based on their scaling proportion.

2. Related Works

Generative models have been widely studied in the field of computer vision. There are two types of generative models: the Variational Auto-Encoder (VAE) [25] and the Generative Adversarial Network (GAN) [17]. The VAE consists of an encoder and decoder. The encoder takes data points from the database and outputs a vector which is perturbed with the Gaussian noise to encourage smooth and complete latent space. The GAN learns a data distribution by finding a Nash equilibrium [16] between the generator and discriminator: the generator is optimized to fool the discriminator and the discriminator is optimized to distinguish the data from the model distribution or the data distribution.

Generative models which synthesize 3D objects with

object priors use an encoder-decoder, for deforming or reconstructing 3D objects. Achlioptas*et al*. [2] proposed the encoder-decoder structure model, which learns point cloud representation by minimizing Earth Movers Distance and Chamfer Distance with the supervision of objects class information. Umetani *et al*. [48] synthesized deformed quad meshes from reference objects by exploring the manifold of the parameterized mesh surfaces with an encoder-decoder framework. Xie *et al*. [53] introduced an energy-based model which approximates the 3D shape probability distribution with Markov Chain Monte Carlo methods such as Langevin dynamics. Groueix *et al*. [18] generate the surface of 3D shapes with a generative model from point cloud objects or images.

The works from this section require reference objects to synthesize or deform objects. However, our proposed model synthesizes objects without observing objects or images as prior information.

Generative models which synthesize 3D objects without object priors are explored as GAN [22, 32], VAE [12, 34], and optimizing latent vectors [5, 13, 36]. 3D-GAN [50] proposed a generative adversarial loss with 3D volumetric convolution and synthesized novel 3D objects from the normal distribution. 3D-IWGAN [42] improved 3D-GAN with Wasserstein GAN [3], which enhances the stability of the learning process. The conditional generative adversarial network (CGAN) [32] generate targeted images given a one-hot encoded class vector. Text2Shape [7] generates colored 3D objects in voxel grids given shape descriptions by jointly learning representations of the text description and 3D colored shapes with metric learning. Conditional Variational Auto-Encoder (CVAE) [24] has been used to learn specific patterns which are structured in the underlying data distribution. CVAE-GAN[4] combines the CVAE framework and adversarial loss to perform fine-grained image generation. The conditional generative model for 3D objects has not been well-explored, and thus, existing methods in the 3D domain have no explicit controllability to sample a specific class of 3D objects with a single pipeline. For example, existing methods require sixteen distinct pipelines to generate objects from sixteen different targeted categories. Our method only requires a single pipeline to generate those objects from all sixteen targeted categories.

3D shape representations learning is used for shape retrieval and reconstruction. Furuya *et al*. [14] proposed Rotation Normalized Grids (RNGs) which are samples of oriented point sets rotated by PCA for shape retrieval. Multiple blocks of RNGs were converted into local features with 3D convolution, and these features are aggregated with average pooling as object representations. Kanezaki *et al*. [21] extracted 3D shape representations by observing multi-view images of an object and jointly estimating their poses. Their method successively works with object classification and shape retrieval tasks. Cohen *et al*. [9] proposed a method that projects 3D views into a sphere by using projections and extracts features from the projected 3D views for the shape retrieval. Dai *et al*. [11] reconstructed a corrupted distance field and state voxel grid with an encoder-decoder model and a shape database. The final output of this method is the reconstructed distance fields of 3D objects in the 3D space. Sung *et al*. [44] used the structure of 3D objects as prior knowledge for shape completion given noisy depth scans of objects. These works were specifically dedicated to tackle shape retrieval or shape reconstruction.

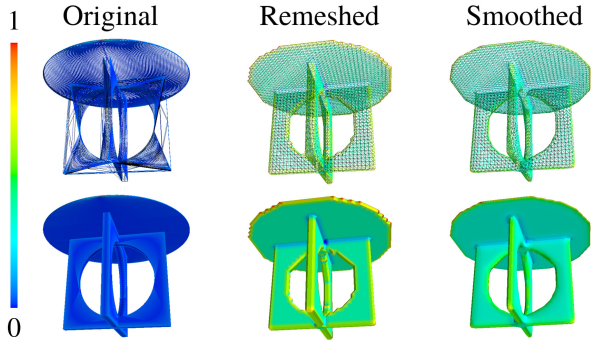

Figure 3: Aligning volumetric space and triangle mesh. Comparing an original mesh with an isosurface from a voxelized object. The first column shows mean curvatures from an original mesh. The second column shows a mesh generated by marching-cubes algorithm given a voxelized object in 40^3 grids. The third column shows the smoothed mesh by applying Laplacian smoothing with the second column mesh.

3. Data preparation

The goal of expanding the dataset is to capture part geometry effectively such as the centroid, surface area, and volume of parts.

3.1. Dataset expansion

In order to create our dataset, *Expanded*, we expand the Projective dataset [20] which is a subset of the ShapeNet [6]. We first choose parts for each class as listed in the supplementary material. Then, we reduce the number of outliers in the labeled vertices from the Projective with Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Furthermore, we manually filter out the objects which contain the remaining outliers. After filtering out the objects, we adjust the vertices (see the supplementary material for detailed procedures). Subsequently, we filter out objects that no longer look like real objects. For example, we removed an object where the table top surface area is smaller than the surface area of the leg. To complete our dataset, we add the mesh models from the Scal-

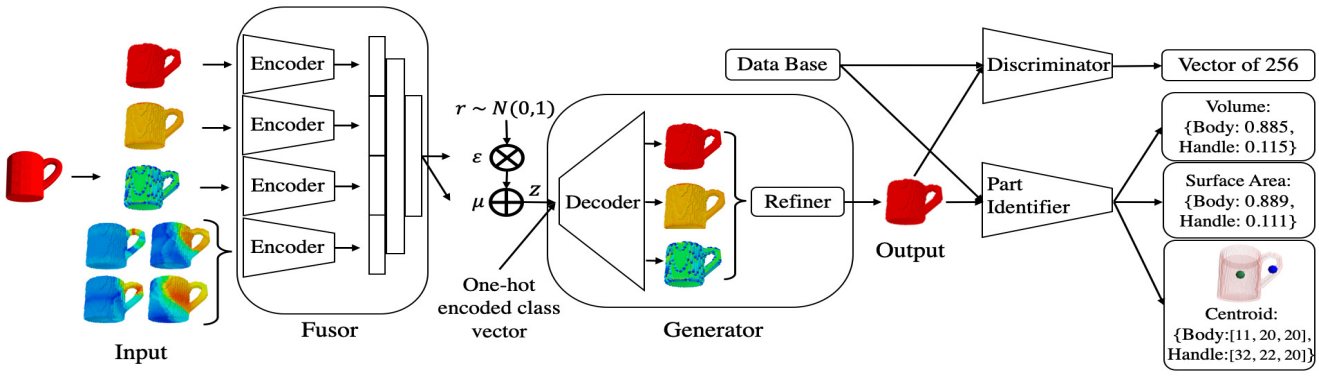

Figure 4: Overview of FuseNet. The fusor consolidates SBMCs, MC, ISV, and IV of the objects and the decoder synthesizes ISV, MC, and IV. A diverse 3D object from a specified category is synthesized given a one-hot encoded class vector and a noise vector from the normal distribution. During the test stage, the fusor is removed from the pipeline. Noise vector **z** is concatenated with a one-hot vector.

able dataset [20] which is a subset of the ShapeNet. The statistics of our dataset are detailed in the supplementary material.

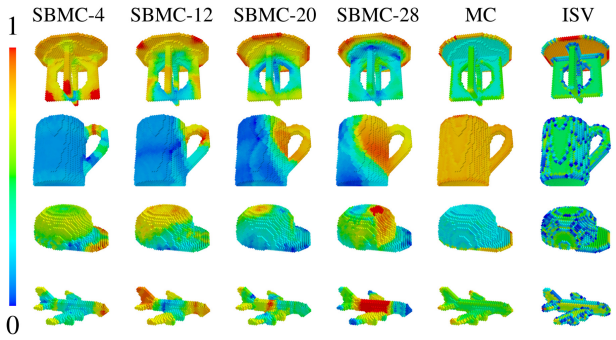

Figure 5: Capturing rapid changes in mean curvatures of local regions. The first, second, third, and fourth columns are SBMC with radius size of two, six, ten, and fourteen voxel units, respectively. The fifth and sixth columns are the conventional MC of a 1-ring neighborhood and ISV, respectively. The resolutions of voxel grids are 40^3.

3.2. Voxelizing surface representation

The meshes are voxelized into a 40^3 voxel grid. Then we use the marching-cubes algorithm [27] to generate triangle mesh since voxelized objects are deformed from the original shape. We further smoothed the meshes with Laplacian smoothing [49]. We extract SBMCs, MC, and ISV into the 40^3 cubic voxel grids. For the ISV extraction, we use SurfaceArea in a library [55] to compute face areas within each cubic voxel. We obtain MC by calculating the average of the minimum and maximum principal curvatures from a 1-ring neighborhood in the triangle mesh. To capture local changes in mean curvature, we use SBMCs (see Figure 5)

Properties	SBMCs	MC	ISV	IV
Interior information				✓
Intersection area			✓	
Rapid local changes	✓			
Association of vertices	✓	✓		

Table 1: Complementary table for three different surface modalities and interior volumetric representations of objects. Rapid local changes are serial information of mean curvature changes in local regions. Intersection area is the intersection area of faces and a cubic voxel.

which is the mean curvature measure of a sphere centered at each vertex in the mesh [10] as expressed in Equation 1. We heuristically choose the radius of the sphere initially to be two voxel units and increase the radius of four voxel units three times (2,6,10,14).

$$\phi^H(B_r) = \int_{B_r \cap S} H(p)\, dp \qquad (1)$$

where S is a surface, and $B_r \subset \mathbb{R}^3$ is a Borel set which is bounded with a sphere of radius r and centered in $p \in S$. $H(p)$ is the mean curvature at point p.

4. Methods

FuseNet consists of four main modules: fusor, generator, part identifier, and discriminator (see Figure 4). We select four different modalities which have complementary properties as summarized in Table 1 for training FuseNet. We integrate ISV, MC, SBMCs, and IV of objects into each encoder as an input and ISV, MC, and IV synthesized as an output from the decoder. Part identifier module learns

part geometry and reduces artifacts of synthesized objects as explored in Section5.2. For inference, FuseNet takes a one-hot encoded vector and a noise vector as an input for synthesizing 3D objects.

4.1. Consolidating modalities and notations

For notations, $\mathcal{D}$, D, G, and I are a decoder, discriminator, generator, and part identifier, respectively. $\odot$ is an element-wise multiplication. The lower index $i \in \{\text{Body, Wheel, ... Legs}\}$ is the index of each part. mc and $\widehat{mc}$ are MC from the object and the decoder, respectively. isv and $\widehat{isv}$ are ISV from the object and the decoder, respectively. pl_i is a $\{x, y, z\}$ Cartesian coordinates of a centroid location of the i th part, pv_i and ps_i are the i th part's volume and surface area, respectively. The $\mathbf{z} \in \mathbb{R}^{200}$ is a vector of $\mu + r \odot exp(\epsilon)$, where $r \sim N(0, 1)$. μ and ϵ are the mean and covariance of the latent vector from the encoder. p_d is data distribution and p_z is a prior distribution of the decoder. $\mathbf{v}$ is the boolean voxels from the true data distribution.

Consolidating both surface and volumetric representations is an informative way to show the geometry of objects. For example, a voxelized volumetric object does not contain curvature explicitly, therefore FuseNet integrates MC and SBMCs to capture curvatures. Our joint representations are complementary (see Table 1), resulting in more robust information than knowing the representation individually. Unlike a volumetric representation alone framework, consolidating complementary knowledge implicitly penalizes the inaccurate estimates. FuseNet is optimized to minimize the $\mathcal{L}_{surface}$ in Equation 2 and thus it learns the curvatures of the objects.

$$\mathcal{L}_{surface} = \sum ||\widehat{isv} - isv||_1 + \lambda \cdot \sum ||\widehat{mc} - mc||_1 \tag{2}$$

Equation 3 and 4 are Sigmoid Cross-Entropy loss. FuseNet learns volumetric representations of objects with the loss term $\mathcal{L}_{volume}$ in Equation 5.

$$\mathcal{L}^{Dec}_{volume} = -\mathbf{v}\log(\mathcal{D}(\mathbf{z})) - (1-\mathbf{v})\log(1-\mathcal{D}(\mathbf{z})) \tag{3}$$
$$\mathcal{L}^{Gen}_{volume} = -\mathbf{v}\log(G(\mathbf{z})) - (1-\mathbf{v})\log(1-G(\mathbf{z})) \tag{4}$$
$$\mathcal{L}_{volume} = \mathcal{L}^{Dec}_{volume} + \mathcal{L}^{Gen}_{volume} \tag{5}$$

In terms of the learning strategy, synthesizing ISV, MC, and IV with a single decoder is multi-task learning. Also, this is done by minimizing $\mathcal{L}_{recon}$ in Equation 6 with a single decoder structure, and thus the model fuses the knowledge of surface and volumetric representations for learning 3D shape representations.

$$\mathcal{L}_{recon} = \mathcal{L}_{volume} + \kappa \cdot \mathcal{L}_{surface} \tag{6}$$

4.2. Learning part geometry

Part geometry is an important factor for understanding shape space [44]. The part identifier module learns part geometry. For training, we provide Cartesian coordinates of centroids, surface areas, and volumes of functional parts by minimizing $\mathcal{L}_p$ in Equation 7. Figure 6 indicates that there are surface area and volume trends between each part. To be more specific, there is a linear slope trend for the volumetric ratio of "Legs" to "Top" of table. Efficacy of part identifier is studied in Section 5.2.

$$\mathcal{L}_p = \sum_i ||\widehat{pl_i} - pl_i||_1 + ||\widehat{ps_i} - ps_i||_1 + ||\widehat{pv_i} - pv_i||_1 \tag{7}$$

The position of the centroid, surface area, and the volume of each part are normalized by dividing the resolution of voxel grids, the surface area, and volume of the object, respectively. The part identifier implicitly learns the relationship between different parts and part geometry by estimating the locations, volumes and surface areas of parts. This module guides the generator towards lower local/global minima of the objective function by minimizing the $\mathcal{L}_p$ with boolean voxels from the data distribution and the model distribution.

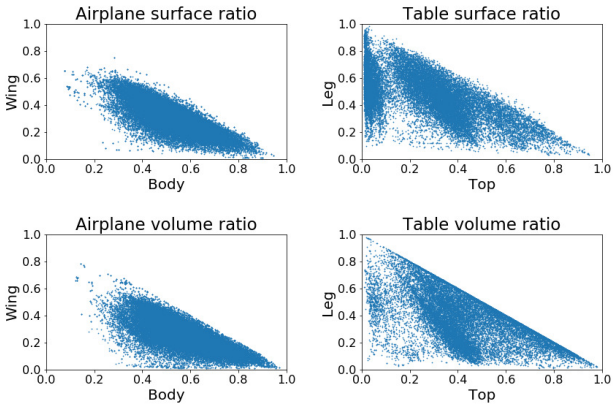

Figure 6: The scatter plots from *Expanded* show that there exists strong relationships within each part. We normalized the volume and surface area of each part with the total volume and surface area of each object, respectively.

4.3. Multi-task learning and loss functions

Many researchers [28, 52, 56] used multi-task learning [47] to prevent overfitting. In the same vein, we adopt a single decoder to share surface and volumetric representation of objects during multi-task learning. Meanwhile, the part identifier module in FuseNet estimates part geometry and the gradients which contain part geometry information backpropagate to the other modules.

To generate synthesized objects of high fidelity, we add least square adversarial loss [30] $\mathcal{L}_D$ and $\mathcal{L}_G$. We use $\mathcal{L}_{kl}$ to minimize the gap between $p(\mathbf{z})$ and the 3D shape distribution in the latent space [4]. Equation 11 shows the final objective function for optimizing FuseNet.

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{v}\sim p_d}[(D(\mathbf{v})-1)^2] + \mathbb{E}_{\mathbf{z}\sim p_z}[(D(G(\mathbf{z})))^2] \quad (8)$$

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{z}\sim p_z}[(D(G(\mathbf{z})-1))^2] \quad (9)$$

$$\mathcal{L}_{kl} = (\mu^T\mu + sum(exp(\epsilon) - \epsilon - 1)) \quad (10)$$

$$\mathcal{L}_{total} = \alpha\cdot\mathcal{L}_G + \beta\cdot\mathcal{L}_p + \gamma\cdot\mathcal{L}_{recon} + \mathcal{L}_{kl} \quad (11)$$

4.4. Training details

To capture the shape representations effectively, we design our encoder by incorporating the Squeeze-and-Excitation block [19] and 3D convolution collaborative filters [45]. To reduce computation and memory consumption, we gradually decode the vector into 2D and 3D by using three 2D convolution layers rather than directly decoding with a 3D convolution layer. We adopt the least square loss for adversarial loss from LS-GAN [30] which stabilizes the training process. We apply the batch normalization to all layers before the activation layer.

For the discriminator, we use Leaky-ReLU as the activation function. We use Swish activation function [38] for other networks. In order to improve the stability of the training procedure, we update the generator twice for each discriminator update to enhance the stability of the learning process [16]. The networks are trained using the ADAM optimizer [23] with the initial parameters: learning rate = 0.0025, β_1 = 0.9, and β_2 = 0.999. For the hyper-parameters, we use $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 0.0001$, κ = 1, and $\gamma = 0.1$. We normalize all data to the interval $[-1, 1]$. The network's architectural details are in the supplementary document.

5. Experiments

We conduct evaluations using 3D shape synthesis, object retrieval, and object reconstruction to validate the efficacy of FuseNet. In general, we divide our dataset into three sets: 70% training set, 10% validation set, and 20% as a test set. We use a single TITAN Xp GPU and an Intel i7-6850K CPU for all of our experiments.

5.1. Evaluation metrics

For the evaluation metric, we use L_1 and Jensen-Shannon Divergence (JSD) [29] for the quantitative measurement. For L_1 calculation, we first binarize the voxels with threshold 0.5 and calculate L_1 distance. JSD is used to measure the similarity between P_v and $P_{\hat{v}}$. To convert boolean cubic voxels in voxel grids to be a probability distribution, we normalize voxels by the total number of occupied voxels. P_v and $P_{\hat{v}}$ are the probability distributions for objects in the database and objects synthesized from the model, respectively. JSD is fundamentally different from L_1 error since it measures the similarity of two occupied voxel distributions.

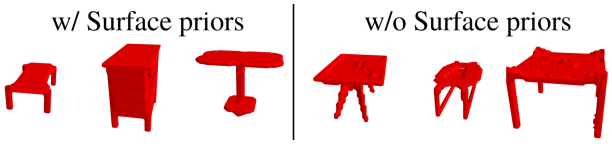

Figure 7: *Grouped* (left) consolidates modalities that generate smoother surfaces than results from *Baseline* (right) which is the volumetric-alone framework.

5.2. Ablation study

Why does fusing surface and volumetric lead to descriptive shape representations? We demonstrate the rationale for fusing surface and volumetric representations of each object (see Figure 7). For the ablation studies, we define three different baselines: (i) *Baseline*, which only uses volumetric representations of objects for training; (ii) *Grouped* consolidates SBMCs, ISV, MC, and IV as inputs and consists of a single decoder to synthesize ISV, MC, and IV; (iii) *Divided* is the same as *Grouped*, but with a different decoder structure which consists of three different decoders where each decoder synthesizes MC, ISV, and IV, separately

We train all the baselines with *Expanded* dataset. Table 2 shows the performance of all baselines. *Baseline* is significantly enhanced by fusing SBMCs, ISV, MC, and IV as used in *Divided* (see Figure 8). Grouping the decoder, as done in *Grouped*, achieves a lower JSD and L_1 than *Divided*. The results indicate that the network learns a structural correlation across the local surface and volume descriptors to improve the fidelity of final outputs. This ablation study validates the rationale for fusing surface and volume as opposed to other choices.

Why learn the geometry of parts? Furthermore, we explore the efficacy of learning part geometry (see Figure 9). Table 2 shows the quantitative evaluation of the proposed method for the following baselines: (i) *w/o Expanded*, which is the same pipeline as *Baseline* but is trained with the Scalable dataset; (ii) *w/o Part identifier* does not have the part identifier in the pipeline; (iii) *w/o SBMCs*, which is trained without SBMCs to evaluate the efficacy of using local changes in mean curvature. The fact that *Baseline* performs better than the *w/o Expanded* validates the efficacy of *Expanded* dataset. Moreover, we verify that $part\ identifier$ improves the quality of synthesized objects since *FuseNet* of L_1 and JSD are lower than *w/o Part identifier*. Also, FuseNet gives lower metric scores than

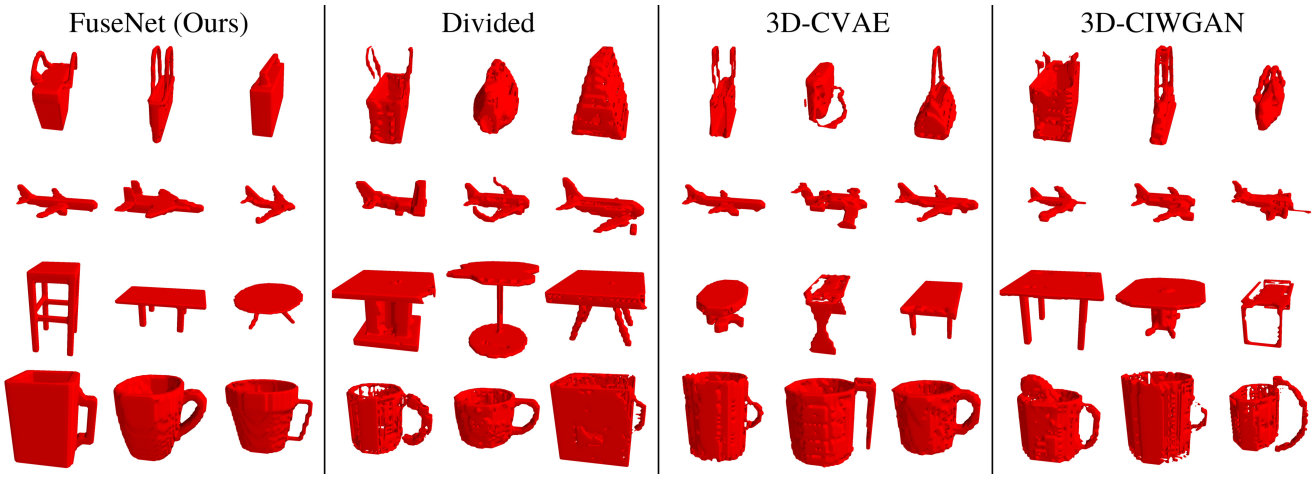

Figure 8: Objects generated from a one-hot encoded class vector and a vector from the normal distribution without a reference object. The resolution of the voxel grids are 40^3 and objects are binarized by 0.5 threshold.

w/o SBMC, which directly shows that adding SBMCs results in learning robust shape representations. As a conclusion, FuseNet performs best among the other baselines, and each of our proposed methods reduces artifacts and holes of synthesized objects.

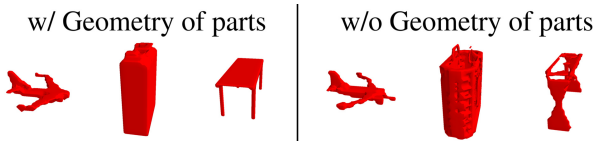

Figure 9: The model using the geometry of parts from FuseNet (left) preserves part properties better than not using the geometry of parts from $w/o\ Part\ identifier$ (right).

	Seen		Unseen	
	L_1	JSD	L_1	JSD
3D-CVAE [54]	0.143	0.008	0.219	0.012
3D-CIWGAN [42]	0.225	0.017	0.312	0.021
Baseline	0.204	0.037	0.252	0.102
Divided	0.143	0.023	0.172	0.042
Grouped	0.111	0.014	0.132	0.018
w/o SBMCs	0.094	0.011	0.118	0.016
w/o Expanded	0.224	0.043	0.312	0.123
w/o Part identifier	0.112	0.012	0.105	0.010
FuseNet (Ours)	**0.052**	**0.003**	**0.082**	**0.005**

Table 2: Quantitative results of baselines and other methods for 3D object generation with 3D model references.

5.3. Synthesizing 3D objects

We conditionally generate 3D objects by sampling a latent vector **z** which was mapped to the object space and a one-hot encoded class vector. We compare our method with 3D-CVAE [54] and 3D-CIWGAN [42]. For 3D-CVAE, we follow CVAE base model in [54] and use the same encoder/decoder definitions as FuseNet. We combine 3D-IWGAN [42] and CGAN [32] for 3D-CIWGAN as a fair comparison. All methods use *Expanded* dataset for training and provide a one-hot encoded class vector as a condition. It takes around seven days to train each model with batch size 16. Figure 8 shows the synthesized objects from our method and others. Since we are sampling objects without referencing them and sampling from noise distribution, it is impossible to display the exact same objects for each method. Each voxel value of the synthesized objects is binarized with a threshold of 0.5. We visualize volumetric data after applying the marching-cubes algorithm and Laplacian smoothing. We observe fewer artifacts and holes in the objects from FuseNet than the others. 3D-CIWGAN is not able to synthesize the nose of the airplane and 3D-CVAE is effective on airplanes but has many holes in the table class. Unlike others, FuseNet preserves part geometry and well-defined surfaces.

We evaluate the feature maps learnt by FuseNet which performs a 3D object classification on the ModelNet [51] in an unsupervised manner [50]. ModelNet has two types of datasets: ModelNet10 and ModelNet40. ModelNet 10 and ModelNet40 consists of ten classes and forty classes, respectively. For a fair comparison with Achlioptas *et al.* [1], which used 57,000 human-made objects from 55 categories, we fine-tune FuseNet without the part identifier module since we do not have the parts' labels in those objects. Af-

	ModelNet10	ModelNet40
VConv-DAE [40]	80.5%	75.5%
3DD [53]	92.4%	N/A
3D-GAN [50]	91.0%	83.3%
GM Point Clouds [1]	95.4%	84.5%
FuseNet (Ours)	**96.1**%	**85.3**%

Table 3: 3D object classification accuracy with unsupervised learning on ModelNet dataset.

ter fine-tuning, we extract features from the last three layers of the fusor module and the second, third, and fourth layers of discriminator. Then, we concatenate features by applying max pooling with the kernel size {8,4,2}, respectively, to create a feature vector and use the same method as in Xie *et al*. [53] for calculating the classification accuracy. Our method outperforms other existing works as shown in Table3. This results indicated that FuseNet learns meaningful shape representations.

5.4. Applications

We evaluate our learnt shape representations by performing shape retrieval and object reconstruction.

5.4.1 3D Object reconstruction

We conduct 3D object reconstruction tasks and compare to 3D-EPN [11]. We used a corrupted dataset from 3D-EPN for training and testing with five classes: Airplane, Car, Chair, Lamp, and Table classes. To feed corrupted objects in 32^3 voxel grids, we modify the fusor module in FuseNet to be consist of an encoder. For the quantitative evaluation, we convert reconstructed objects into boolean voxels with a threshold of 0.5. Then we counted wrongly estimated voxels and divided by total number of occupied voxels in ground truth. We use pre-trained weights of 3D-EPN which use class information from the 3D-EPN project page. The error of FuseNet and 3D-EPN are reported in Table 4. Qualitative results are displayed in Figure 10.

Class (# of train objects)	3D-EPN [11]	FuseNet (Ours)
Airplane (3.3K)	0.226	**0.202**
Car (5K)	0.197	**0.191**
Chair (5K)	0.309	**0.273**
Lamp (1.8K)	0.407	**0.392**
Table (5K)	0.338	**0.251**
Total (20.1K)	0.286	**0.249**

Table 4: Quantitative results of 3D-EPN [11] and FuseNet.

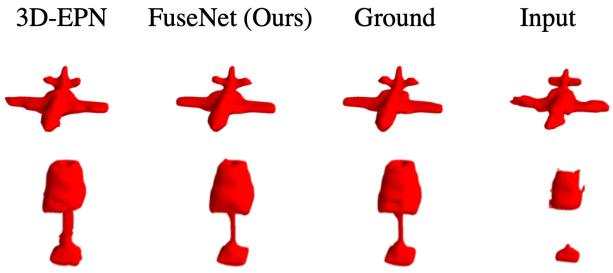

Figure 10: Qualitative results from 3D-EPN and FuseNet.

5.4.2 3D object retrieval

We perform a shape retrieval on ShapeNet Core55 dataset [39], following the rules of the SHRECH17 3D shape contest. In our experiment, we use the aligned dataset to test our learned shape descriptors from FuseNet. The encoder of the learned model from the shape synthesis experiment is transfered and fine-tuned for the feature extraction. We use the output of the final (softmax) layer for measuring distances of each object [21]. For the evaluation, the final results are calculated with the official metrics from the trained descriptors. We list the top four, and we compare our method with other methods as shown in Table 5.

	P@N	R@N	F1@N	mAP	NDCG@N
A[21]	0.810	0.801	0.798	0.772	0.865
B[39]	0.786	0.773	0.767	0.722	0.827
C[46]	0.765	0.803	0.772	0.749	0.828
D[14]	0.818	0.689	0.712	0.663	0.762
Ours	0.812	0.776	0.785	0.754	0.831

Table 5: Results and best competing methods for the SHREC17 shape retrieval competition. A, B, C, and D are Kanezaki *et al*. [21], Zhou *et al*. [39], Tatsuma *et al*. [46], and Furuya *et al*. [14], respectively.

6. Conclusions

We propose FuseNet that fuses surface and volumetric representations of objects and uses part geometry for shape analysis and synthesis. Our focus is to explore the performance of learning shape representations by fusing surface modalities and interior volumetric information which has complementary features from the other. FuseNet exceeded state-of-the-art methods in shape synthesis and reconstruction and is on par with state-of-the-art methods in shape retrieval. For future work, fusing other surface modalities with volumetric representations could be further explored with a recurrent neural network in sparse matrix format.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 1, 7, 8

[2] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds, 2018. 1, 3

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 3

[4] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: fine-grained image generation through asymmetric training. *CoRR, abs/1703.10155*, 5, 2017. 3, 6

[5] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017. 3

[6] 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. 1, 3

[7] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018. 1, 3

[8] 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. Springer, 2016. 1

[9] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 3

[10] D. Cohen-Steiner and J.-M. Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321. ACM, 2003. 4

[11] A. Dai, C. Ruizhongtai Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. 1, 3, 8

[12] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth. Learning diverse image colorization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[13] J. Fan and J. Cheng. Matrix completion by deep matrix factorization. *Neural Networks*, 98:34–41, 2018. 3

[14] T. Furuya and R. Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *BMVC*, pages 121–1, 2016. 3, 8

[15] 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. Springer, 2016. 1

[16] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 2, 6

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[18] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Atlasnet: A papier-m\^ ach\'e approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*, 2018. 1, 3

[19] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 6

[20] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3d shape segmentation with projective convolutional networks. In *Proc. CVPR*, volume 1, page 8, 2017. 2, 3, 4

[21] A. Kanezaki, Y. Matsushita, and Y. Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 1, 3, 8

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[24] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014. 3

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[26] A. Kotte, N. Van Wieringen, and J. Lagendijk. Modelling tissue heating with ferromagnetic seeds. *Physics in Medicine & Biology*, 43(1):105, 1998. 1

[27] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of graphics tools*, 8(2):1–15, 2003. 4

[28] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014. 5

[29] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999. 6

[30] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. On the effectiveness of least squares generative adversarial networks. *arXiv preprint arXiv:1712.06391*, 2017. 6

[31] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 1

[32] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3, 7

[33] D. Nelson, S. Charbonnel, A. Curran, E. Marttila, D. Fiala, P. Mason, and J. Ziriax. A high-resolution voxel model for predicting local tissue temperatures in humans subjected to warm and hot environments. *Journal of Biomechanical Engineering*, 131(4):041003, 2009. 1

[34] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016. 3

[35] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3500–3509, 2017. 1

[36] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*, 2019. 3

[37] 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. 1

[38] P. Ramachandran, B. Zoph, and Q. V. Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 2017. 6

[39] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, X. Bai, et al. Shrec16 track: largescale 3d shape retrieval from shapenet core55. In *Proceedings of the eurographics workshop on 3D object retrieval*, pages 89–98, 2016. 8

[40] A. Sharma, O. Grau, and M. Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016. 8

[41] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proc. CVPR*, 2017. 1

[42] E. Smith and D. Meger. Improved adversarial systems for 3d object generation and reconstruction. *arXiv preprint arXiv:1707.09557*, 2017. 3, 7

[43] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[44] M. Sung, V. G. Kim, R. Angst, and L. Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015. 3, 5

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 6

[46] A. Tatsuma and M. Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval. *The Visual Computer*, 25(8):785–804, 2009. 8

[47] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 2, 5

[48] N. Umetani. Exploring generative 3d shapes using autoencoder networks. In *SIGGRAPH Asia 2017 Technical Briefs*, page 24. ACM, 2017. 3

[49] J. Vollmer, R. Mencl, and H. Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum*, volume 18, pages 131–138. Wiley Online Library, 1999. 4

[50] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 3, 7, 8

[51] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 7

[52] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4460–4464. IEEE, 2015. 5

[53] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018. 1, 3, 8

[54] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016. 7

[55] D. Yarotsky. Geometric features for voxel-based surface recognition. *arXiv preprint arXiv:1701.04249*, 2017. 1, 4

[56] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1036–1041. IEEE, 2014. 5