Quad mesh network and its applications

Zhangsihao Yang
Carnegie Mellon University
zhangsiy@andrew.cmu.edu

Abstract

Functionally generating 3D shape is an extremely tough problem. In this work, we use a novel representation of 3D shape which is capable of doing convolution on it and train an autoencoder as well as an info GAN to generate meaningful results. We introduce several processes on the 3D shape domain: 1) propose a new 3D shape representation to encode a 3D shape to an image, 2) we design several loss functions to better train a 3D shape autoencoder, 3) leverage feature based InfoGAN to generate meaningful 3D shape. The results show that this process is helpful with raw 3D shape design and could discover difference inside the unique category. We further analyze the style transformability of our autoencoder network. The result could the smooth effect of our network is capable of discovering underneath essence of 3D shape. By using the discriminator of Info GAN, we could do weakly supervised classification with the same category.

1. Introduction

Detailed analysis of structures in 3D shape is a necessity for numerous computer graphic applications. For example, it is important in the domain of 3D shape to classify and segment a model into the different category and separate part. For generating, there is a need to obtain features of 3D shape and use that features to replicate the same model, while design domain requires automatic methods to generate an unexpected novel shape, structure or other relevant outlines.

Neural network has proven to be an effective way of classification, segmentation and extracting features. It is well known that what kind of data format to be used for 3D shape is an open hot topic. Feeding typically fixed data size into the neural network, however, is an extremely challenging task, given that a 3D shape could have varying size.

In this paper, we present a novel data format for representing 3D shape feature problem to fix the problem mentioned above. That is, we propose a new 3D representation that mapping a 3D mesh to an image with a fixed dimension. Using that representation, the 3D shape could be an image that stores the x, y, and z coordinate information. And the traditional convolution method could be applied to the 3D shape.

In our work, we have achieved to train an autoencoder that is capable of generating a 3D mesh. And a feature-based InfoGAN is trained to generate meaningful 3D mesh features. With these two networks trained, controlling the generated shape is available by changing the input to the genenror of InfoGAN. Meanwhile, we test the style transform of our autoencoder network and the results show that the goal of style transform is archived.

Contributions. Using deep learning to aid design to improve the efficiency of the raw design. Our contributions include:

1. Apply convolution on the quad mesh to extract information on the 2D manifold and use that information to reconstruct the 3D quad mesh.
2. Design a feature-based InfoGAN network that could be used to generate meaningful 3D meshes to satisfy certain design propose
3. Use the smoothing function of the autoencoder network to archive style transformation on the 3D mesh
4. Proved that the discriminator of InfoGAN is capable of doing weak unsupervised classification on models with the same category

2. Related work

3D shape analysis can be considered as a large if not the ultimate domain. To devise our solution, we build upon three sub-fields: deep learning on 3D shape, and generative model.

2.1. Deep learning on 3D shape

A broad area of active research in geometric deep learning (see [2] for a review) is closely related to this paper. The boom of deep learning (in particular, convolutional architectures) in image-field has brought a hot interest in the computer graphics community to replicate this progress for applications dealing with geometric 3D data. One of the key

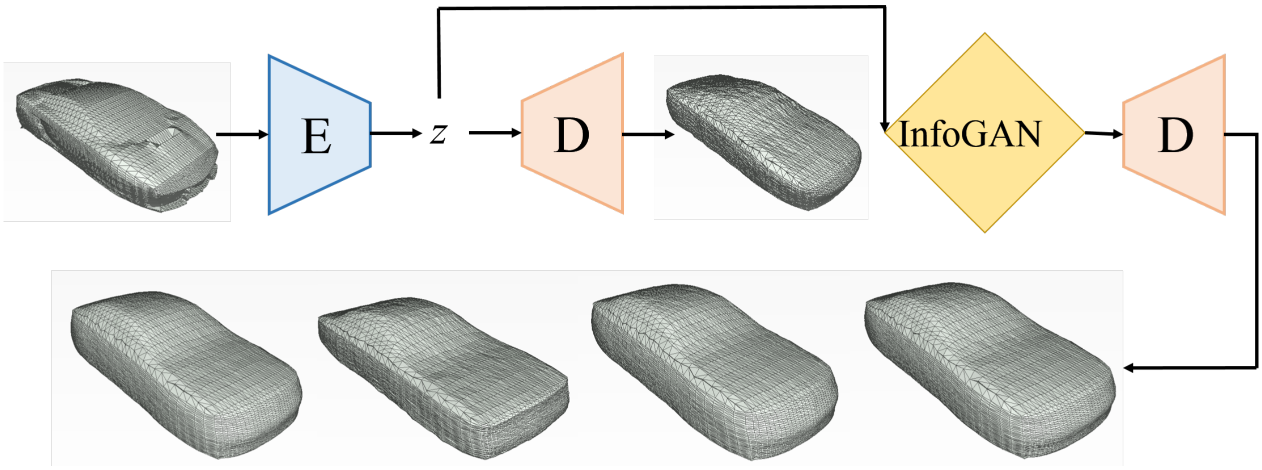


Figure 1. The structure of quad mesh InfoGAN.

difficulties is that for such data it requires great care to define the basic operations constituting deep neural networks, such as convolution and pooling.

Some work avoids this problem by using the Euclidean representation of the 3D shape, such as a collection of point clouds [8], or a volume representation [7]. One of the main drawbacks of this depth learning approach is that they do not do any operation on the real 2D manifold. In addition, voxel representations are usually memory intensive and have poor resolution [7]. The PointNet model [8] applies the same operations to the coordinates of each point and aggregates the local information without allowing interaction between different points, which makes it difficult to capture local surface properties. PointNet ++ [9] solves this problem by proposing a multi-level layering model. But it still does not touch the surface of the 3D model no matter how dense the point could it is.

2.2. Generative model

The state-of-the-art in generative modeling has rapidly advanced with the introduction of Variational Autoencoders (VAE [6]), Generative Adversarial Networks [4]. These advances are both good quality work for 3D shape domain to generate 3D models. The work that direct put hands on the 3D mesh surface is rarely using generative models.

3. Proposed method

In this section, the methods used in our paper will be discussed in details. First, we show a bi-direction mapping that could convert 3D models into a 2D map. After converting, an autoencoder is trained to extract the features of 3D models. Then using the features extracted, an InfoGAN is trained on the features and feed the generated features back to the autoencoder structure to get the generated 3D models. More details are explained below.

3.1. Quad-mesh data set

A data set that has converted all the vehicle into the registered mesh is introduced by Nobuyuki Umetani[10]. For each of the vehicle in this data set, the number of faces is six which is corresponding to the six faces of a cube and fixed. And the subdivision routine for each vehicle is same across different models. For more details about how to generate this data set, you could refer to the original paper. To be more clear, that all the mesh in this data set has the same connection.

3.2. Quad-mesh representation in neural network

As we have seen throughout the past few years that many 3D shape format has been used in deep learning. But quad mesh is seldom used. Several constraints could raise when speaking of the usage of quad-mesh. First, the quad is not very popular in 3D shape domain. Most mesh file is using triangle mesh instead of quad mesh. The reason behind is that generating high-quality quad-mesh is harder. Second, the quad mesh has its own limitations. For example, some shape is much easier to represent with triangle mesh while it will use more quad polygon to represent the same shape by using quad-mesh.

However, because of the shape and high quality of quad mesh, the quad mesh could be a better data format to be used in deep learning problem. We define a way to convert a quad-mesh to an image which popular in deep learning. Most of the neural network's inputs are images because of the boom of convolution on GPU. The overall procedure of converting is illustrated in figure 2. And this procedure is just like peeling an orange. The difference is that for the face that is not surrounded by other faces, we pad the faces that around that face.

By doing the procedure illustrated above, we have converted a 3D shape with six faces into an image then many

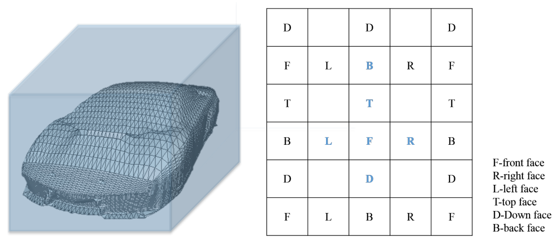


Figure 2. The way to convert a quad mesh to an image. Six faces of a 3D shape are associated with the grid box covered it. And unfold the box from the front face then concatenate the surrounding face with each unfolded faces

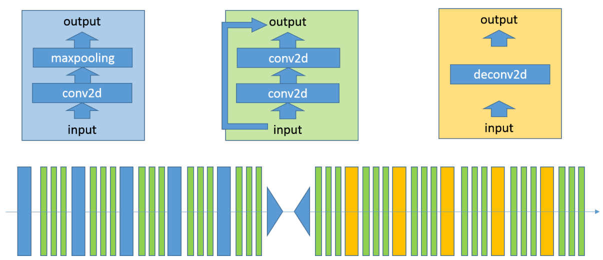


Figure 3. The network architecture of our autoencoder network, the top left module is a downsampling model which is used at the encoder part. The top middle one is the rest new module which is used to increase the depth of the network and help with decreasing reconstruction loss. The top right one is an upsampling module which is used at the decoder part of the network. The structure graph at the bottom is the final version of our autoencoder network

operations that we could do on the 2D domain will be applied on the 3D domain.

3.3. Auto encoder on 3D map

Figure 3 shows the network architecture of 3D map autoencoder. The input to this network is a 3-channel image. The channels are corresponding to x coordinate, y coordinate and, z coordinate. So the convolution operation on the 3D map is like filer swiped on the outline of the 3D shape. The output of this network is the same 3D map as the input. And the height and width of the 3D map are 193 and, 161. Each block in this network is a ResNet[5] module. And the bottleneck size is 1024.

3.4. Generative model

Speaking generative model, VAE[6] and GAN[4] are two mainstream generative model that has been widely used. Especially, GAN has attracted most attention once it is born. The reason of popularity is the min-max game played by generator and discriminator or game theory of generator and discriminator.

In our paper, InfoGAN[3] is used to generative features or 3D model. GAN has its own limitation because of the instability of training and unexpected generator which means that the output of GAN's generator is not controlled by the user. And what user could do is to feed a random noise into the generator.

InfoGAN is aimed to solve the problem above. In the architecture of InfoGAN, the input to the generator is some meaningful vector and this meaningful vector is retrieved from discriminator such that the what is generated from the generator is controlled by the input to the generator. In our paper, InfoGAN's network architecture is using all fully connected layers. The task of the generator is to generate features of 3D shape.

4. Experimental Results

In this section, we provide extensive results of our model. We report how the loss function affects the results of Auto encoder and compare with and without certain loss. We then train an InfoGAN network to find the meaningful information behind all the 3D models. Finally, we evaluate our autoencoder with another task, which is to smooth the original 3D shape and use that shape to do style transform between different 3D model. Note that the feature generated by InfoGAN is passed to decoder part of the autoencoder to get the 3D map of the 3D shape.

4.1. Reconstructing the 3D map

In this section, we describe our full pipeline for extracting features from the 3D map and use that features to reconstruct the 3D map. We implement our quad-mesh autoencoder with Tensorflow[1] and test on a single NVIDIA GTX 1080 Ti GPU.

The reconstruction validation loss is around 0.002393 which is a decent magnitude of loss. And the training loss is around 0.002366 which is not far away from the validation loss. A By comparing training loss and validation loss, we can see that the autoencoder network is not overfitting. Figure 4 shows the whole process of training. And reconstruction and the input 3D map is shown in figure 1 (The left-hand side of the encoder is the input and the right-hand side of the decoder is the output). The results seem to have been smoothed by the network. And it is a common phenomenon happening at training VAE. Because VAE has the potential property to ignore detail on the input images. We will use this property to do style transform at later section 4.3.

And some self-defined loss is critical for the autoencoder to learn how to reconstruct a nice output 3D map. These losses are edge length loss, normal loss, stitching loss and, pointwise MSE (Mean Squared Error) loss.

We have given some comparisons between different loss's effect on reconstruction 3D map. This is also called ablation study. In figure 5, it shows how the result when we add edge length to the loss and when we do not. As it

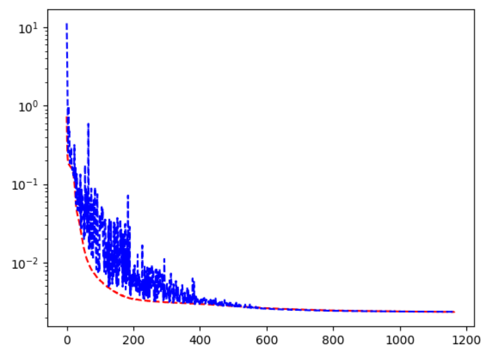


Figure 4. The curve of loss when training the autoencoder network. The blue line is the validation loss curve and the red line is the training loss curve

shows, without the edge length loss some points on the reconstructed mesh would stay far away from the ground truth mesh. And edge length loss could help with this problem by regularizing the point to be close to its neighbor points. By applying edge length loss, the generated mesh could become much closer to its neighbor ring. And the results look more smooth comparing not using it.

The norm loss is a more sophisticated loss which means it trying to discover some latent property of mesh and send that information to the network and tell the network a direction about which way is better to reconstruct the 3D mesh. The norm of a point is computed according to its neighborhood points. And the norm is a higher lever information comparing to the point wise loss. The neural network may find out how to using convolution operation to compute the normal of the point and use that information to compute the loss. But with the experience of engineering, to accelerate the network to discover to utilize norm as a loss, we could directly use the norm difference from input to the output as a loss to modify the parameter of the network. And the results of training with and normal loss is shown in figure 5. And we could see that more details are generated once the norm information is leveraged. And that feature could be learned with a smaller number of training epochs and it is a way to accelerate the training of an autoencoder.

4.2. InfoGAN to generate 3D mesh

We trained an InfoGAN on the features encoded by the encoder network we have obtained. In training info GAN, we used the parameter recommended by the authors of info GAN. The network details are described as follows.

In this network architecture, the target generated 'images' are the feature of the 3D meshes. Here the 'images' are the images generated by traditional GAN. And out results of generating novel 3D shape is shown in figure 6, 7, and 8. From the category aspect, the network itself could discover that the feature of classic cars and sport cars and SUV have a different kind of features, And we could control which kind of cats to be generated by giving different cat information as input to the network by concatenating with continuous information and random noise.

For the aspect of the continuous part, the network could discover how to control the different part of the generated cats. For example, the first continuous noise could be used to control the back top of the car. When the value input to the network is increasing, the top of that cat could become higher and higher. And the second continuous noise could control the width and the heights of a vehicle together which makes sense in reality. Because there is some design consideration in the car industry when you increase the width of the car normally the height of the cat will decrease. And this property is also shown in our result. You could see that the changing happening at the outline of the 3D shape in figure 8.

One thing that is interesting of InfoGAN is that, it could help with classification on one category as it is shown in figure 8. For example in our experiment, the discriminator could help with distinguishing different cart in the data pool. To be more specific, the discriminator could extract information from the features of 3D meshes. And using that information we could know which category this car belongs to. And also what kind of continuous features this car probably contains. So once the InfoGAN is trained, it could not only generate new unseen shapes. It could also be helpful with classification and segmentation meaning interpolation.

4.3. Using the reconstruction results to do style transformation

As a byproduct of our autoencoder network, the reconstruction results of our autoencoder network have a smooth function. The output of the autoencoder network is a smoothed version of the input 3D shape. And we could use that property to get the smooth 3D shape. The reconstruction 3D shape could be regarded as the 'bone' of the mesh. When we subtract the mesh with the 'bone' of the mesh, the 'skin' of the mesh could be obtained. So you could imagine that different cars have different bones and skins. But you could change the skin of a car to another car's skin. Thus the style of the car changes from one style to another style.

As it is shown in figure 9, the style transformation between two different kinds of the car is pretty clear. The transformation happens between a sporty car and an old-fashion car. By doing subtraction, we could obtain the 'skin' of a sporty car and old-fashion car. And adding that 'skin' to different cars, the style of that car could be changed with that 'skin'.

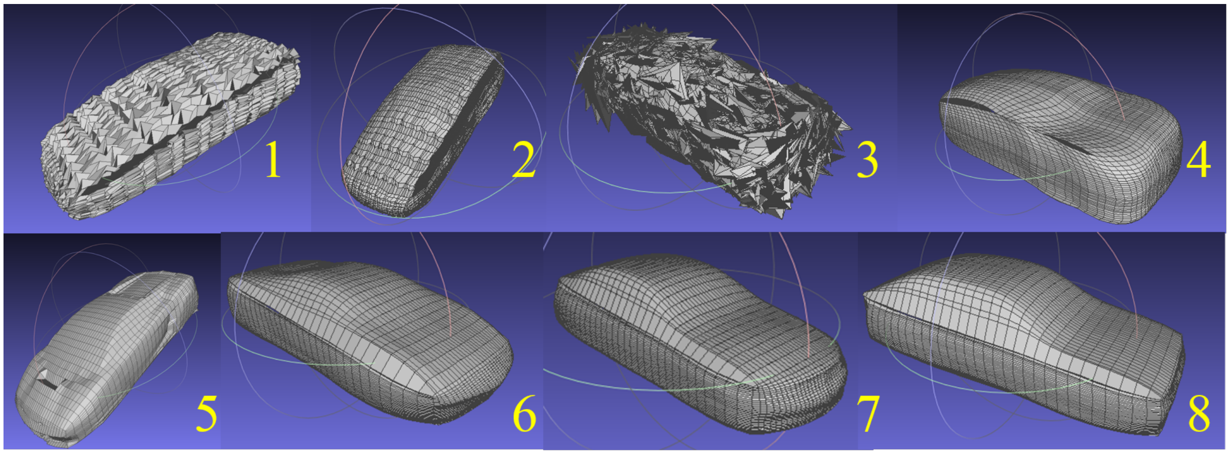


Figure 5. Ablation study on the loss function. 1: the reconstruction result with only pointwise MSE loss; 2: the reconstruction result with pointwise MSE loss and edge length loss; 3: the reconstruction result with only pointwise MSE loss; 4: the reconstruction result with pointwise MSE loss, normal loss and edge length loss; 5: the ground truth input to the network; 6: The reconstruction result with all the losses; 7: The reconstruction result with all the losses; 8: The reconstruction result with all the losses

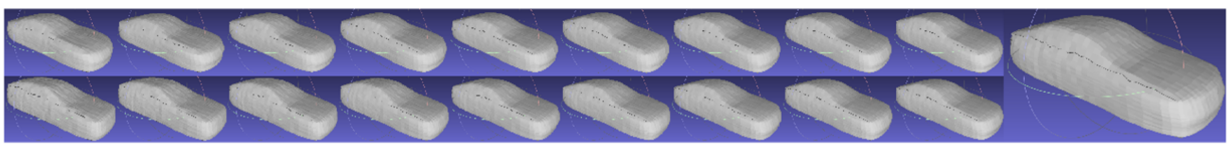


Figure 6. The change of 3D shape when the first continuous value varying. From the top left to the right the continuous value increasing from -2.0 to 0.0 with step 0.05. And from the right to bottom left the continuous value input to the generator changes from 0.0 to 2.0 with step 0.05

5. Conclusion

In this paper, we proposed quad mesh info GAN, a model for generating the meaningful 3D shape that can be used to assist interactively design. The model builds on an autoencoder network structure and introduces two losses in 3D mesh domain which could be helpful with future exploration on 3D mesh reconstruction. We further show that info GAN could generate desired category, contiguous property and features of 3D shape. We also explore the side effect of our model to smooth mesh model as well as to distinguish different style in the same domain, and show style transforms between different models.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. 3

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *CoRR*, abs/1611.08097, 2016. 1

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. 3

[4] 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, 3

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 3

[7] S. Liu, A. G. O. II, and C. L. Giles. Learning a hierarchical latent-variable model of voxelized 3d shapes. *CoRR*, abs/1705.05994, 2017. 2

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation.

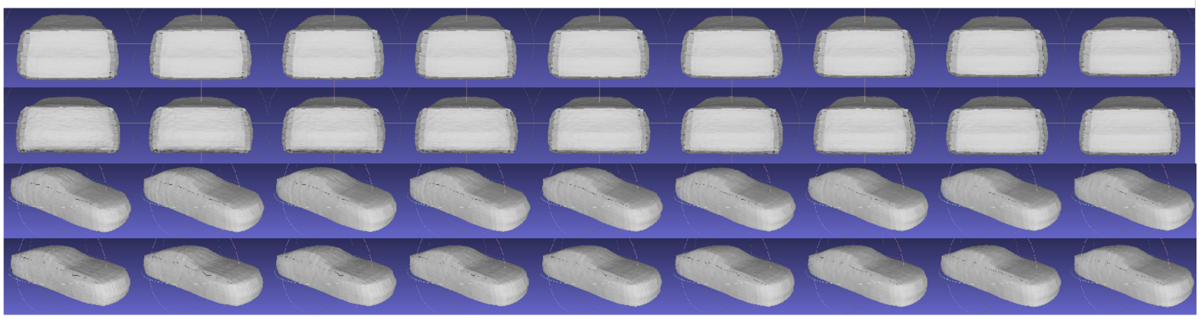
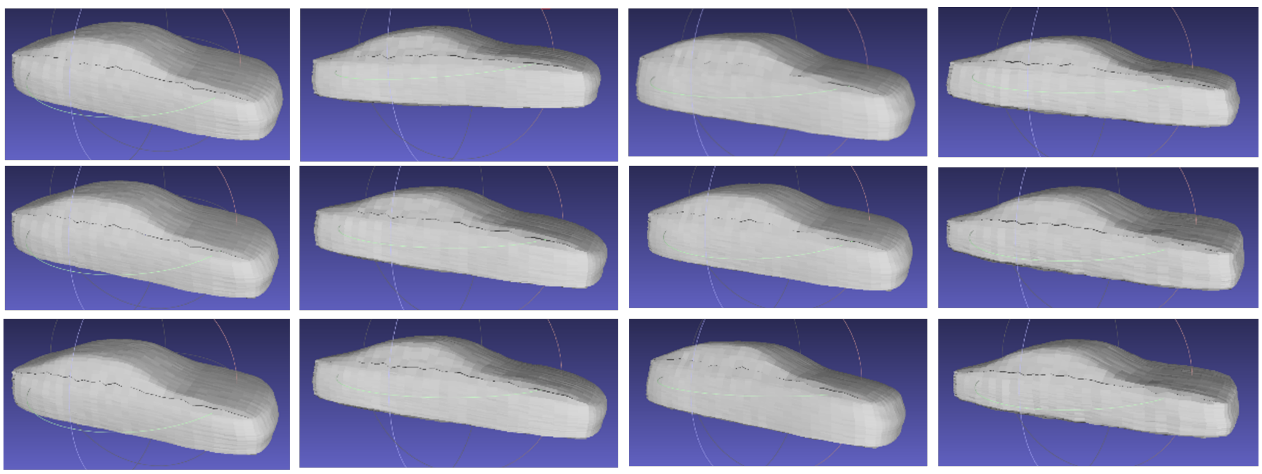


Figure 7. The change of 3D shape when the second continuous value varying. From the top left to the right the continuous value increasing from -2.0 to 0.0 with step 0.05. And from the right to bottom left the continuous value input to the generator changes from 0.0 to 2.0 with step 0.05



C1 = 0 C1 = 1 C1 = 4 C1 = 8

Figure 8. The change of 3D shape when the category one-hot vector varying. From left to right, each one corresponding to one category in the InfoGAN

CoRR, abs/1612.00593, 2016. 2

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 2

[10] N. Umetani. Exploring generative 3d shapes using autoencoder networks. In *SIGGRAPH Asia 2017 Technical Briefs*, SA '17, pages 24:1–24:4, New York, NY, USA, 2017. ACM. 2

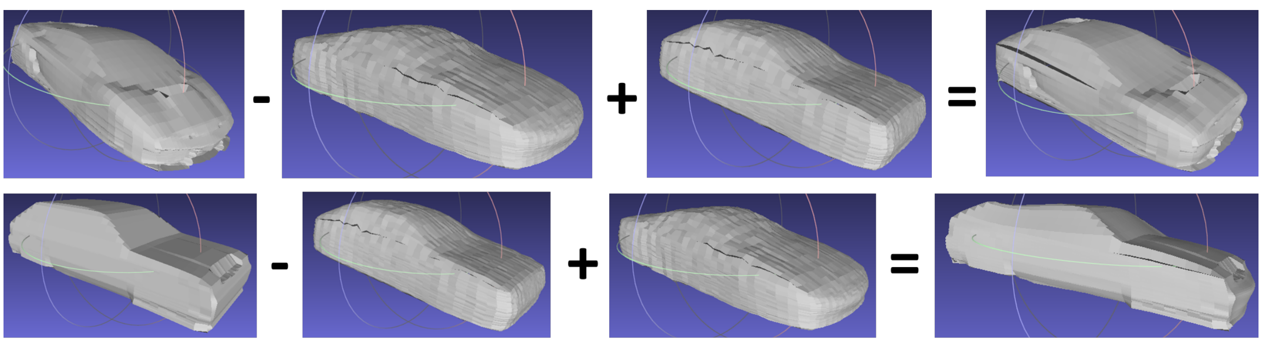


Figure 9. By doing simple arithmetic, the style could be transformed from one vehicle to another vehicle. For the first row and the second row, from left to right are the initial 3D meshes, the smoothed 3D meshes, the target 'bone' shapes, and the transformed results