Single view 3D object reconstruction

Bastien DECHAMPS
MVA
bastien.dechamps@eleves.enpc.fr

Mathieu ORHAN
MVA
mathieu.orhan@eleves.enpc.fr

Abstract

In this work, we study two methods allowing to directly recover a 3D mesh from a 2D image. We compare their quantitative and qualitative results with similar preprocessing and dataset, and investigate the effect of two mesh regularization schemes.

1. Introduction

Single view 3D object reconstruction (SVR) aims at recovering a 3D representation from an image. It is a central challenge in computer vision and is clearly ill-posed. Among possible output representations, we focus on meshes. Using directly meshes is very advantageous compared to other representations, such as voxels or point clouds. In computer graphics, it is the representation of choice: sparse, memory-efficient, and easy to manipulate. While most of the literature focus on other representation, some of the very recent approaches [2, 5, 7] directly manipulates meshes.

All of these methods are based on deep learning, thus are optimizing an objective. This objective is generally regularized to produce meshes of better quality. In this work, we study the effect of different regularizers introduced in [7] on two of the state-of-the-art methods, namely AtlasNet [2] and Pixel2Mesh [7]. Our objectives are the followings: (1) using common data, preprocessing, and metrics, compare the two methods quantitatively and qualitatively, (2) experiment with regularization schemes on the two methods.

2. Methods

In this section, we briefly summarize how [2, 7] are applied to the SVR task, and detail the mesh regularization schemes.

2.1. AtlasNet

Overview AtlasNet seeks to approximate the target surface by mapping a set of squares in 2D to the surface of the 3D shape using Multi-Layer Perceptrons (MLPs). These learnable parametrizations transform 2D templates (e.g., unit squares) into smooth 2-manifolds in 3D that will cover the surface of the object. Inducing a mesh on these 2D templates will naturally transfer it on the target surface.

Image features extraction AtlasNet uses a ResNet architecture [4] to encode the 2D image of an object into a latent vector. This vector is then given to multiple MLPs with points sampled from the templates to form the corresponding point cloud located on the surface of the 3D representation.

Losses AtlasNet optimizes the Chamfer distance between the obtained point cloud and the ground truth. Given $S_1, S_2 \subset \mathbb{R}^3$, the Chamfer distance $d(S_1, S_2)$ is defined as:

$$d(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \tag{1}$$

This distance only takes point clouds as input and does not directly reflect the quality of an induced mesh, in opposition to the next studied method, where mesh regularizers are added to the loss.

2.2. Pixel2Mesh

Overview Pixel2Mesh seeks to progressively deform an initial ellipsoid mesh in a coarse-to-fine fashion. This is enabled by Resnet-like [3] Graph Neural Networks (GNN), which use image features extracted with a 2D Convolutional Neural Network (CNN).

Image feature extraction A VGG-16 [6] architecture extracts features from multiple stages and uses them as additional vertex features. To merge them, the 3D mesh is projected onto the image.

Graph Convolutional Neural Network A deformation block update vertex features, which include in particular

the 3D coordinates of the vertex. It is composed of a succession of simple message passing steps organized in residual groups. An unpooling layer increases the number of vertices in input mesh. It allows to progressively recover the details. The whole GNN is composed of 3 deformation blocks, and 2 unpooling layers.

Losses Pixel2Mesh uses the Chamfer loss to enforce point cloud reconstruction, and a normal loss to ensure normal consistency. The authors propose two regularizers, essential in their method. They are introduced in the next section.

2.3. Regularization

To prevent the optimization process to be stuck in a local minima, a mesh regularization term is added to the loss. In this report, we focus on the two particular regularization types introduced by Pixel2Mesh: *Laplacian regularization* and *Edge Length regularization*.

Laplacian regularization This regularizer allows neighboring vertices in the mesh to have the same movement during optimization and avoids mesh self intersection. It is defined by the following:

$$l_{\text{lap}} = \sum_{p} \|\delta'_p - \delta_p\|_2^2 \tag{2}$$

where $\delta_p = p - \frac{1}{|\mathcal{N}(p)|} \sum_{k \in \mathcal{N}(p)} k$ is the Laplacian coordinate of p and δ'_p is the same quantity after deformation.

Edge-length regularization This regularizer prevents outlier vertices in the mesh by penalizing the length of edges:

$$l_{\text{edge}} = \sum_{p} \sum_{k \in \mathcal{N}(p)} \|p - k\|_2^2 \tag{3}$$

In Pixel2Mesh, these losses are computed block-wise, that is a loss term is added for each deformation block. Adapting these regularizers to AtlasNet is not straightforward, as we will discuss in the section 3.3.

3. Experiments

3.1. Data and preprocessing

We benchmarked the two methods on the ShapeNetv1 [1] dataset. We limit our experiments to the *cars* category, to alleviate the training time with our limited computational resources. These methods both use this dataset, but with different preprocessings. For fair comparison, we use the exact same version of ShapeNet for both methods, and the same train and evaluation splits. As Pixel2Mesh requires camera intrinsics, we chose to use their version of ShapeNet. We started to work from official public authors implementations publicly available on github (https://github.com/ThibaultGROUEIX/AtlasNet and https://github.com/nywang16/Pixel2Mesh). We use MeshLab and Visdom to visualize the produced point clouds and meshes.

Object views are resized to a dimension of 224×224 pixels, and we kept the original processing of the methods except that. Surprisingly, Pixel2Mesh's authors did not normalize their point clouds. This is mandatory to compare the methods, as the Chamfer distance is not invariant to scaling. We chose to use the unit ball normalization in both methods. We did try to train AtlasNet without normalization, but noticed poor performances and instability. Finally, we sampled the point cloud to have 2500 points, as AtlasNet struggles to deal with more points.

3.2. Performance comparison

Following the same preprocessing, we trained both methods on the same data with different hyperparameters. On table 1, the Chamfer loss as well as the F1-score are compared between the different models. On the car dataset,

	Chamfer	F1-score
Atlas (1 templ.)	5.11	0.41
Atlas (5 templ.)	4.93	0.43
Atlas (25 templ.)	4.95	0.42
P2M	**3.98**	**0.54**

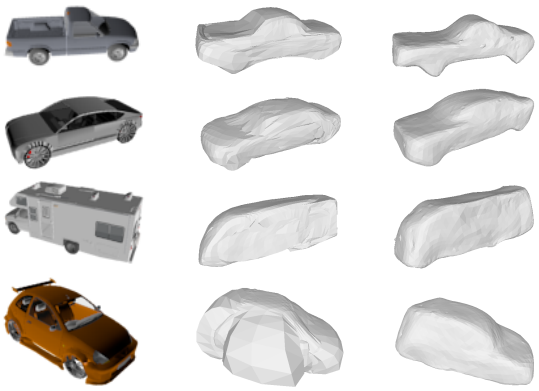
Table 1: Performance of both methods (cars)

Pixel2Mesh seems to outperform by far AtlasNet on both the Chamfer distance and the F1-score, even by increasing the number of templates used for reconstructing the surface in AtlasNet. Furthermore, we observe that using 25 templates tampers the results, which is not expected and goes in opposition to the results presented in the original paper.

On figure 1 are presented some qualitative results comparing the meshes generated by both methods. We can clearly observe that the meshes produced by Pixel2Mesh are much smoother than those from AtlasNet. This is due to the regularization terms that operate directly on the mesh, while AtlasNet does not directly optimize meshes. AtlasNet meshes often contain some outliers as on the fourth example of figure 1.

3.3. Regularization

Ablation study We first conduct an ablation study on Pixel2Mesh to observe the effect of the two regularizers presented before, and reported some results in the figure 2. As expected, without edge-length regularization, the resulted meshes contain edges whose length is very irregular, as well as holes in it. Without Laplacian regularization, the resulted



(a) input image (b) AtlasNet (c) Pixel2Mesh

Figure 1: Generated meshes from both methods (zoom for details).

meshes lose their smoothness. However, the model trained without edge-loss has the best performances as shown in table 2. This is not surprising as the Chamfer distance and the F1-score do not reflect mesh quality but rather are used for comparing point clouds. Furthermore, ground truth point clouds not only contain points on the surface of the 3D object but also inside it. This can explain the hole on the car meshes on figure 2b coupled with those good performances.

	Chamfer	F1-score
P2M (with both)	3.98	0.54
P2M (- edge-length)	**3.76**	**0.55**
P2M (- Laplacian)	3.95	0.53

Table 2: Results of the ablation study for the two regularization schemes on Pixel2Mesh

(a) with both (b) - Edge-length (c) - Laplacian

Figure 2: Some meshes produced by Pixel2Mesh with and without regularization (zoom for details).

AtlasNet regularization AtlasNet does not consider any mesh regularization schemes and only optimizes the Chamfer loss, which does not directly reflect mesh quality. Moreover, except during inference and validation time, it does not even manipulate a mesh but only point clouds. To improve the produced meshes, we added a regularization term to the Chamfer loss:

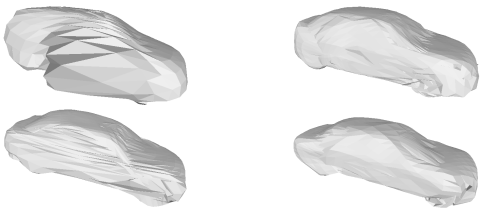
$$\mathcal{L}(S_\theta, S_{\text{gt}}) = \mathcal{L}_{\text{chamfer}}(S_\theta, S_{\text{gt}}) + \lambda \sum_{k=1}^{K} l_{\text{edge}}(\mathcal{M}_\theta^k) \quad (4)$$

where S_θ is the generated point cloud, S_{gt} the ground-truth point cloud, K the number of templates and $\mathcal{M}_\theta^k$ the produced mesh for the k-th template. The latters are obtained by performing a Delaunay triangulation of the sampled points on each template. We focused on edge-length regularization and did not consider the Laplacian regularizer.

	Chamfer	F1-score
Atlas (5)	4.93	0.42
Atlas (5) + reg	**4.89**	**0.43**

Table 3: AtlasNet, regularization effect.

We trained two models with 5 templates, with and without regularization. Quantitative results are reported on table 3 as well as example of produced mesh on figure 3. Not only regularization improves the performances but it also remove some of the outliers present on figure 3. However, it is still far from the results obtained with Pixel2Mesh.



(a) no reg. (b) reg.

Figure 3: Some meshes produced by AtlasNet with and without regularization (zoom for details).

4. Conclusion

In our experiments, Pixel2Mesh outperforms AtlasNet both qualitatively and quantitatively by a large margin. The outputs of the latter lacked smoothness and were prone to outliers. However, our results suggest that regularizers directly operating on meshes can improve the results of AtlasNet. Furthermore, some of our experiments prove that the Chamfer distance is not the best suited distance if one wants to operate on meshes.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and

Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. cite arxiv:1512.03012. 2

[2] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. 1

[5] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 1

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 1

[7] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 1