Cerberus: A Multi-headed Derenderer

Boyang Deng, Simon Kornblith, and Geoffrey Hinton
Google Brain
Toronto
{bydeng,skornblith,geoffhinton}@google.com

Abstract

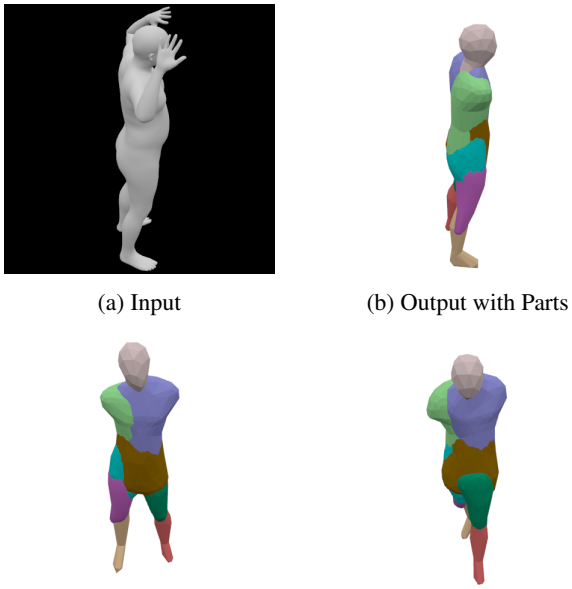
To generalize to novel visual scenes with new viewpoints and new object poses, a visual system needs representations of the shapes of the parts of an object that are invariant to changes in viewpoint or pose. 3D graphics representations disentangle visual factors such as viewpoints and lighting from object structure in a natural way. It is possible to learn to invert the process that converts 3D graphics representations into 2D images, provided the 3D graphics representations are available as labels. When only the unlabeled images are available, however, learning to derender is much harder. We consider a simple model which is just a set of free-floating parts. Each part has its own relation to the camera and its own triangular mesh which can be deformed to model the shape of the part. At test time, a neural network looks at a single image and extracts the shapes of the parts and their relations to the camera. Each part can be viewed as one head of a multi-headed derenderer. During training, the extracted parts are used as input to a differentiable 3D renderer and the reconstruction error is backpropagated to train the neural net. We make the learning task easier by encouraging the deformations of the part meshes to be invariant to changes in viewpoint and invariant to the changes in the relative positions of the parts that occur when the pose of an articulated body changes.

Cerberus, our multi-headed derenderer, outperforms previous methods for extracting 3D parts from single images without part annotations, and it does quite well at extracting natural parts of human figures.

1. Introduction

In this work, we present Cerberus, a neural network that extracts a part-based 3D graphics representation from a single image, along with a training strategy that avoids the need for part annotations by using natural consistencies, *i.e.* the invariance of part shapes under changes in viewpoint or pose.

This training strategy ensures that Cerberus can learn to reconstruct geometrically correct 3D graphics models consisting of semantic parts without part supervision. The arrangements of the 3D parts extracted by Cerberus change with pose, and we can manipulate the 3D model to form a novel pose (Figure 1d). We examine Cerberus on two datasets of articulated bodies. On the human dataset, which has substantial variability in pose, Cerberus not only outperforms previous work by a large margin, but also learns semantic parts such as head and legs without part annotations. These parts are consistent across poses; Cerberus produces better results than baselines even when it is restricted to applying parts extracted from an image of an individual to all other images.



(a) Input (b) Output with Parts

(c) New Viewpoint (d) New Pose

Figure 1: Given an input image (a), Cerberus can output a part-based 3D model of the object (b). With this 3D model, we can look at the object from a new viewpoint (c) or manipulate the parts to generate a new pose (d).

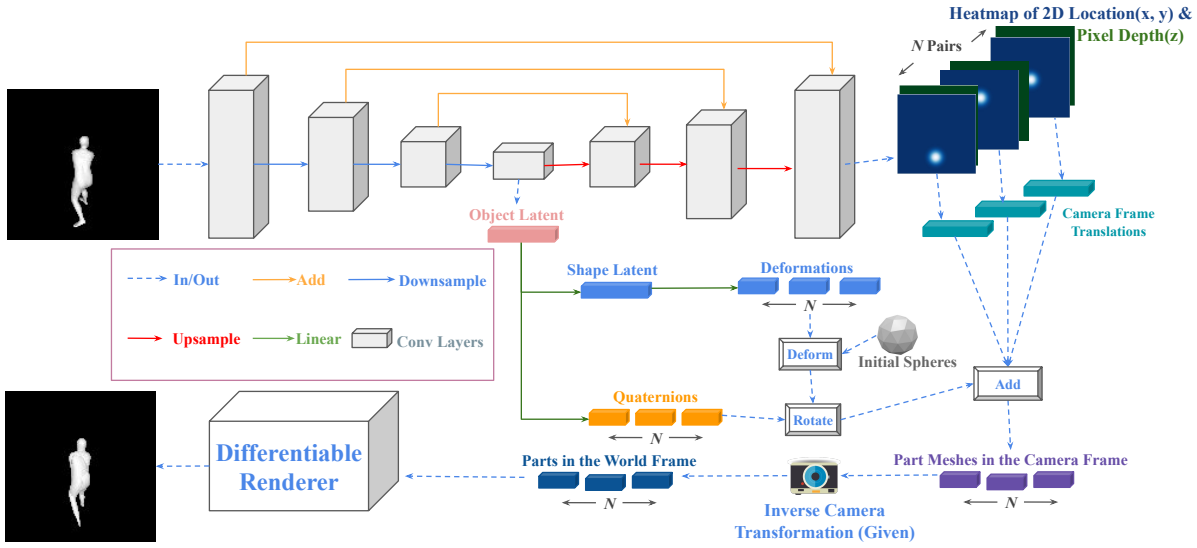


Figure 2: Cerberus architecture. Here we visualize 3 out of N parts used in the pipeline. Up-sampling is performed by de-convolution. The object latent is obtained by global average pooling of the lowest-resolution feature maps. Predicted quaternions are used to construct rotation matrices.

2. Cerberus Architecture

2.1. 3D Parameterization

We describe 3D shapes by deforming a spherical mesh [1]. The edges, faces, and initial positions of all the vertices of this sphere are predefined. To model the shape of a part, the neural network predicts the displacement of each vertex.

We use an independent mesh for each part. We parameterize a part's local pose by its rotation and translation relative to the camera. The neural network predicts parameters of these transformations based on the pose of the object in the input image. After applying transformations to each part and putting them together in the same 3D space, we obtain a 3D model of the whole object in a specific pose.

2.2. 3D Reconstruction Pipeline

Our pipeline is illustrated in Figure 2. Given an input image, our pipeline outputs the 3D parameters defined in Section 2.1 for all the parts. The number of parts, N, is a predefined hyper-parameter. As shown in Figure 2, we use a base network similar to the hourglass block [3] to extract deformation, rotation, and translation parameters from a single image. For the translation of each part, we linearly transform the feature maps and apply a spatial softmax, yielding a "probability map" $\{p^k_{x,y}\}$. We compute the 2D coordinates for the part by taking the expectation over this map. We also calculate a depth map $\{d^k_{x,y}\}$ with elements represent the depth of pixel (x, y) for the k-th part.

The resulting translation T_k for the k-th part is:

$$T_k = \pi^{-1}\left(\sum_{(x,y)\in G}[x \cdot p^k_{x,y}, y \cdot p^k_{x,y}, d^k_{x,y} \cdot p^k_{x,y}]\right) \quad (1)$$

During testing, the produced 3D model is our output. During training, rather than employing 3D supervision, we use a differentiable renderer to transform 3D representations into images. We render our 3D representation, compare the rendered result, R, with the input image, I, and then back-propagate through the renderer. The objective we use here is mean squared error pixel reconstruction loss:

$$\mathcal{L}_r = \frac{1}{|G|}\sum_{(x,y)\in G}(I_{x,y} - R_{x,y})^2 \quad (2)$$

3. Consistency Constraints

3.1. Pose Consistency

Although we learn part-based models to reconstruct 3D objects, we do not use any part supervision or keypoint annotations during training. Instead, we reflect on the way humans split an articulated body into multiple parts and find an essential hint on how to split semantic parts without supervision: For a pair of images of the same person in 2 poses, the 2 predicted sets of parts should have the same shape. In practice, we use a pair of images from the same viewpoint containing the same object in 2 different poses for training. We argue that collecting this kind of supervision is trivial, since we can simply use 2 frames from a video of a moving object filmed by a static camera.

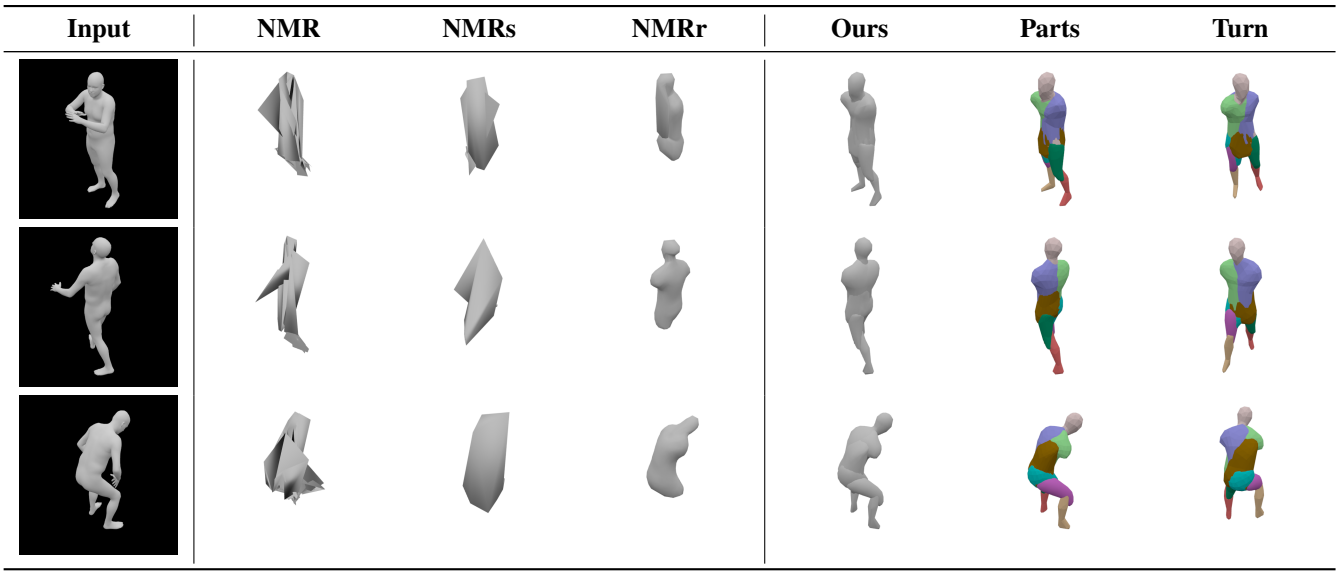


Table 1: 3D Human Model Reconstructions. On the left are input images. In the middle are baseline results. NMR is the architecture and implementation from [1] without a smoothness loss. NMRs adds smoothness loss to NMR. NMRr is our re-implementation of NMR using the same renderer and images with shading as used by Cerberus. We also visualize our parts in different colors in the middle column of the right group. The Turn column shows outputs rendered from a new viewpoint.

Model	Human	Human Hard	Animal
NMR	0.2596	-	0.3000
NMRs	0.2233	-	0.2574
NMRr	0.3084	-	0.3201
Cerberus	0.4970	**0.4728**	**0.4255**
Free Cerberus	**0.5099**	0.4365	0.4196

Table 2: Single image 3D reconstruction test results on 2 datasets. NMR refers the model proposed by [1]. NMRs is NMR with smoothness loss. NMRr is our re-implementation using the same renderer as Cerberus. Free Cerberus is Cerberus trained without pose consistency. Human Hard reflects accuracy when reconstructing all poses in the test set using the same set of parts. NMR models can not do this test due to its lack of pose variance.

3.2. Viewpoint Consistency

Learning 3D shape from a single image is an ill-posed problem. There exist an infinite number of possible 3D models that yield the same 2D projection. For the sake of learning correct shapes, we need predicted 3D models to be consistent across viewpoints during training. Specifically, we use a pair of images from 2 different viewpoints for the same object (in the same pose) during training. The goal is to predict the same 3D model from these 2 viewpoints

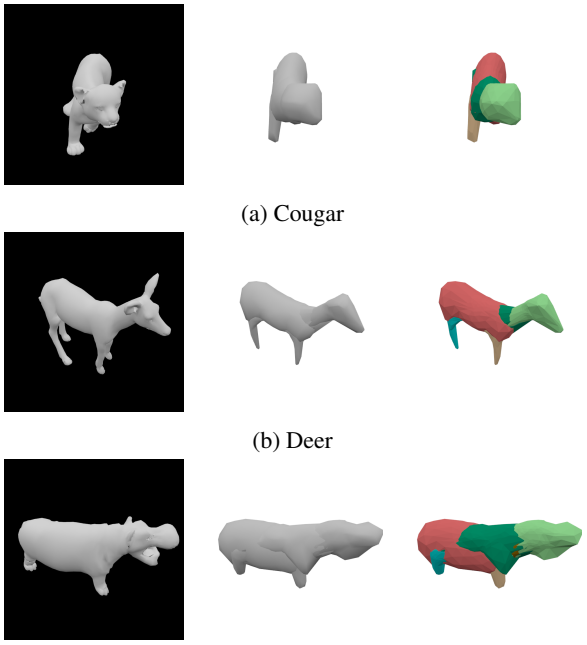
4. Experiments

We test Cerberus on 2 datasets: Human and Animal. The Human dataset contains 3D human models in diverse body poses. We use SMPL [2], a parameterized deformable human model, to generate all the example meshes. The training set comprises 19,500 pairs of body poses with 5 subjects. For the test set, we use 2 unseen subjects. We render 810 different poses of them, each from 4 viewpoints. The Animal dataset consists of 3D models of quadrupeds. Compared with the Human dataset, it has more variance in shape but less variance in pose. Each example is generated by a deformable model, SMAL [5], for quadrupeds. This dataset contains 38,540 training pose pairs and 984 test examples.

4.1. 3D Human Reconstruction

We first test our model on single image 3D human reconstruction. We measure the quality of the predicted 3D models by voxel IoU (intersection-over-union) with voxel resolution of 32^3, following [4]. We provide baseline results of the Neural Mesh 3D Renderer (NMR) [1] both with and without its smoothness loss. To exclude confounds related to the choice of renderer, we also re-implement NMR using our renderer on images rendered with shading. We visualize some example 3D outputs of all the baselines and Cerberus on the test set in Table 1.

As shown in Table 1, Cerberus predicts smooth 3D meshes that are visually more similar to the human in the input. Compared with NMR, which mainly reconstructs the



(a) Cougar

(b) Deer

(c) Hippo

Figure 3: 3D Animal Reconstruction. We visualize 4 examples from the test set and the 3D outputs of Cerberus. **Left:** Input images. **Middle:** 3D outputs. **Right:** Parts rendered in different colors.

outline shape of the torso, Cerberus can produces more details of the body, *e.g.* the legs. We hypothesize that this improvement is related to the flexibility of part-based modeling. More importantly, we find that, although our model is trained without any part annotations, it can predict semantic parts of the human body.

Quantitatively, Cerberus predicts 3D meshes with greater similarity to the target meshes than previous approaches (shown in Table 2). Compared with the original NMR, Cerberus achieves double the test IoU. Our re-implementation of NMR has a higher accuracy than the original NMR, suggesting that reconstructing shaded images helps learn better shapes. Nonetheless, Cerberus outperforms this re-implemented NMR by a substantial margin.

We also perform a quantiative evaluation on the learned parts. Instead of computing IoU when reconstructing each test case independently, we perform identity-conditional reconstruction. We first extract the deformed part meshes from 2 images of the 2 subjects in the test set. These images contain the canonical pose of the the subjects. Then, we reconstruct other examples in the test set by applying predicted rotation and translation to the parts from the canonical pose of the same subject. The voxel IoU of this more challenging evaluation ("Human Hard") is shown in Table 2.

4.2. 3D Animal Reconstruction

In addition to reconstructing 3D human models from a single image, we also examine Cerberus's capability of reconstructing objects with greater variability in shape. To this end, we evaluate our method on the animal dataset. We show test IoU on the animal dataset in Table 2. Baseline methods perform better on the animal dataset as compared to the human dataset, likely because there is less variability in pose. Nonetheless, Cerberus remains superior. Thus, Cerberus consistently outperforms baselines on objects with either high variability in pose (humans) or high variability in shape (animals). We demonstrate the predicted 3D animals and parts from Cerberus in Figure 3. We see that Cerberus predicts high quality meshes for different animals in various poses from diverse viewpoints. Additionally, part segmentation is reasonable and consistent across different animals, but parts vary appropriately in shape.

5. Conclusion

We have proposed a new architecture and training paradigm for single-image 3D reconstruction with only 2D supervision. Our approach not only reconstructs 3D models more accurately than approaches that use a single monolithic mesh, but also infers semantic parts without part-level supervision. Although we focus on the problem of 3D reconstruction, in the spirit of inverse graphics, our approach can potentially be adapted to tasks such as classification or pose estimation.

References

[1] 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. 2, 3

[2] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015. 3

[3] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 2

[4] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 3

[5] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6365–6373, 2017. 3