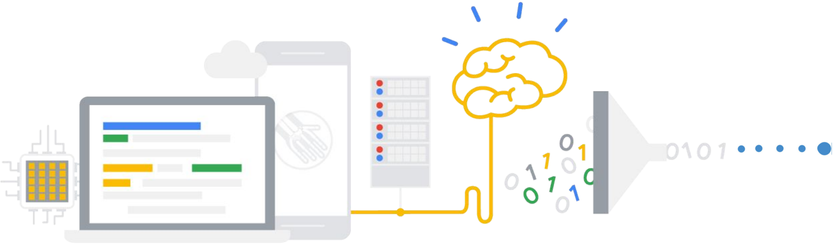
Cerberus: A Multi-Headed Derenderer

Boyang Deng, Simon Kornblith, and Geoffrey Hinton

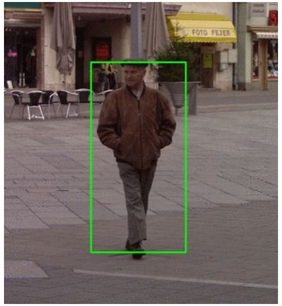
Google Brain, Toronto

Google AI



Modeling objects in images

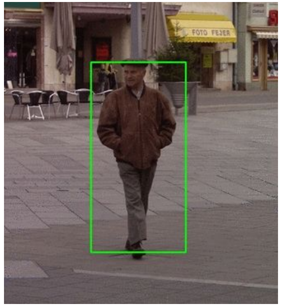
1. 2D Representation?



Modeling objects in images

1. 2D Representation?

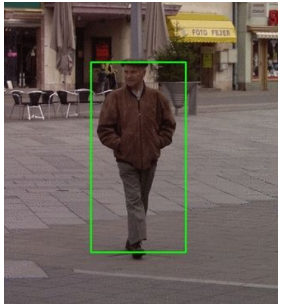
- Not enough, because a lot of objects, e.g. animals, are inherently 3D.



Modeling objects in images

1. 2D Representation?

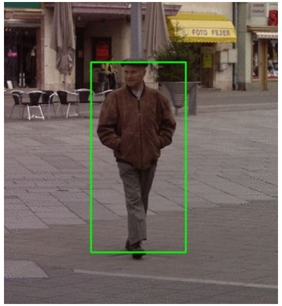
- Not enough, because a lot of objects, e.g. animals, are inherently 3D.
- 2D means a hard time for **novel viewpoints** and **novel lightings**, especially in generation.



Modeling objects in images

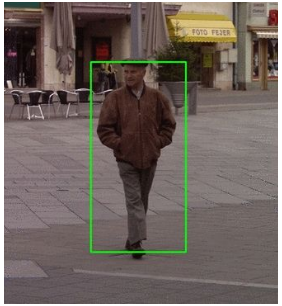
1. 2D Representation?

- Not enough, because a lot of objects, e.g. animals, are inherently 3D.
- 2D means a hard time for **novel viewpoints** and **novel lightings**, especially in generation.
- And …… this is a 3D Vision workshop.



Modeling objects in images

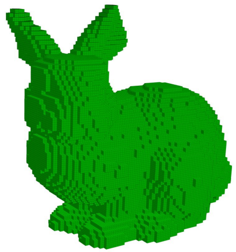
1. 2D Representation? Not Enough.

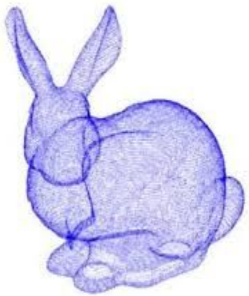


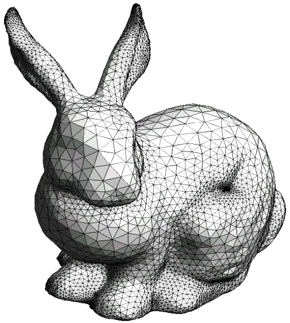
Modeling objects in images

1. 2D Representation? Not Enough.

2. 3D Representation?





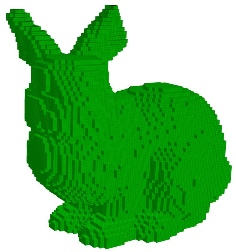


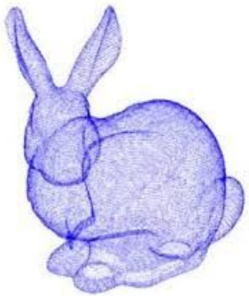
Modeling objects in images

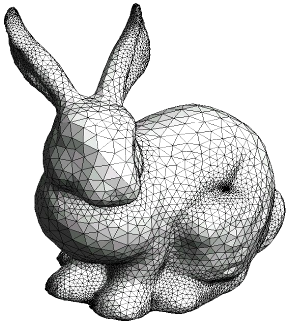
1. 2D Representation? Not Enough.

2. 3D Representation?

- Good. Representing objects as they are.



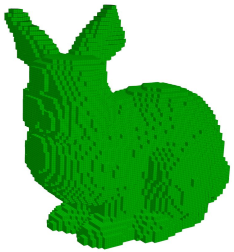


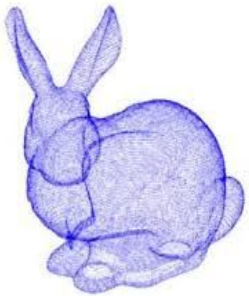


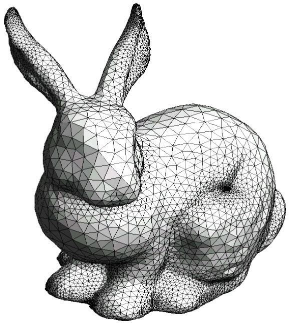
Modeling objects in images

1. 2D Representation? Not Enough.
2. 3D Representation?

- Good. Representing objects as they are.
- Friendly to re-rendering under novel scenes. Thanks to Graphics.





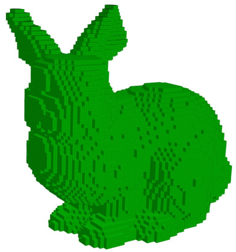


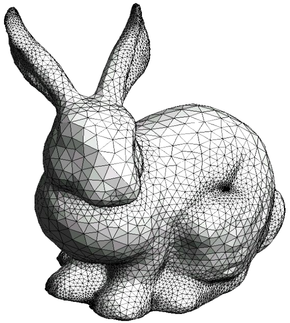
Modeling objects in images

1. 2D Representation? Not Enough.

2. 3D Representation?

- Good. Representing objects as they are.
- Friendly to re-rendering under novel scenes. Thanks to Graphics.
- Which one to use? Voxel, point cloud, or mesh.





Modeling objects in images

1. 2D Representation? Not Enough.
2. 3D Representation?

- We choose mesh.
    - Compact compared to voxel.
    - Fast rasterization -> Plug a differentiable renderer in training -> <u>**No 3D supervision**</u>.

Modeling object in images

1. 2D Representation? Not Enough.

2. 3D Representation? Mesh.

3. Mesh for Articulated Bodies

- Previous methods[1, 2] use a **single mesh** with fixed topology.
- Articulated bodies have poses, i.e. relative locations and orientations of parts.

[1] Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[2] Wang, Nanyang, et al. "Pixel2mesh: Generating 3d mesh models from single rgb images." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Modeling object in images

1. 2D Representation? Not Enough.
2. 3D Representation? Mesh.
3. Mesh for Articulated Bodies

- Previous methods[1, 2] use a **single mesh** with fixed topology.
- Articulated bodies have poses, i.e. relative locations and orientations of parts.
    - Single mesh is hard to fit various poses.

[1] Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[2] Wang, Nanyang, et al. "Pixel2mesh: Generating 3d mesh models from single rgb images." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

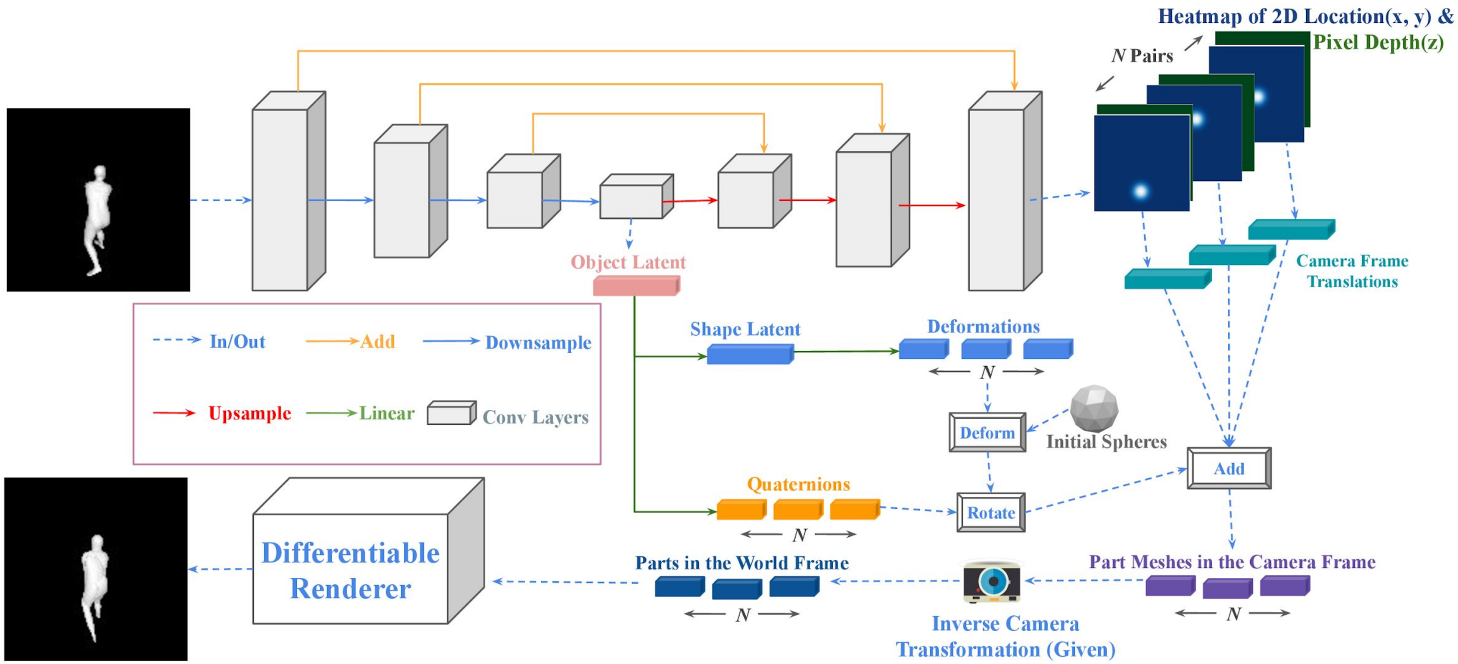
Modeling object in images

1. 2D Representation? Not Enough.
2. 3D Representation? Mesh.
3. Mesh for Articulated Bodies

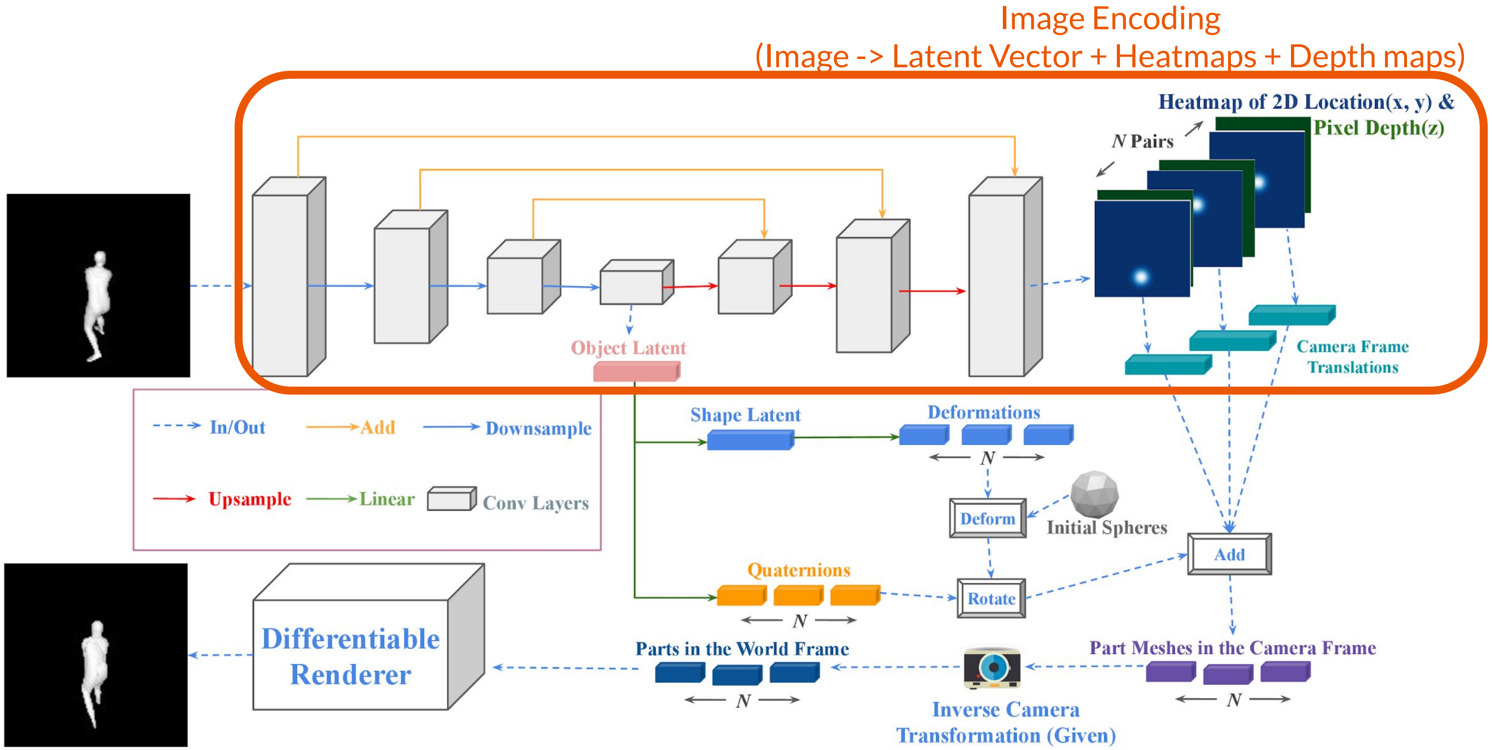
- Previous methods[1, 2] use a **single mesh** with fixed topology.
- Articulated bodies have poses, i.e. relative locations and orientations of parts.
  - Single mesh is hard to fit various poses.
  - Solution: **A Part-based Model**.

[1] Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[2] Wang, Nanyang, et al. "Pixel2mesh: Generating 3d mesh models from single rgb images." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

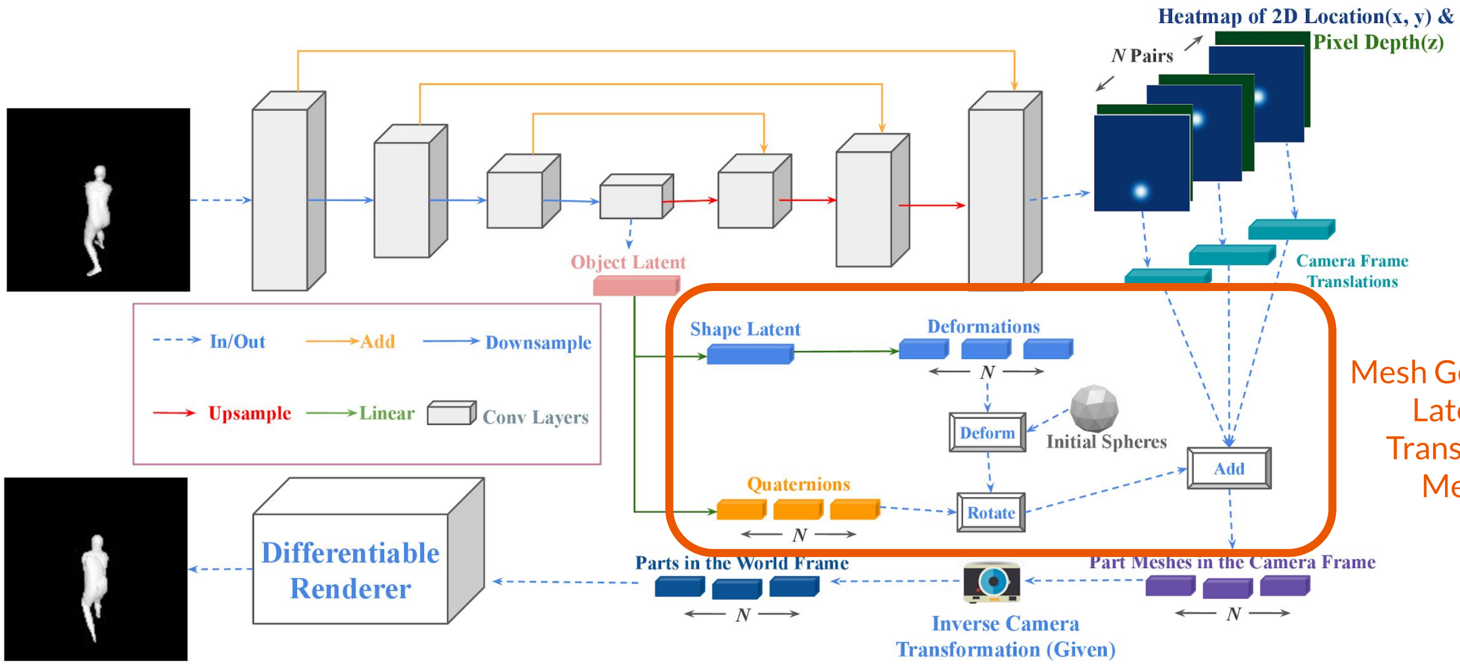
Cerberus: A Multi-headed Derenderer



Cerberus: A Multi-headed Derenderer

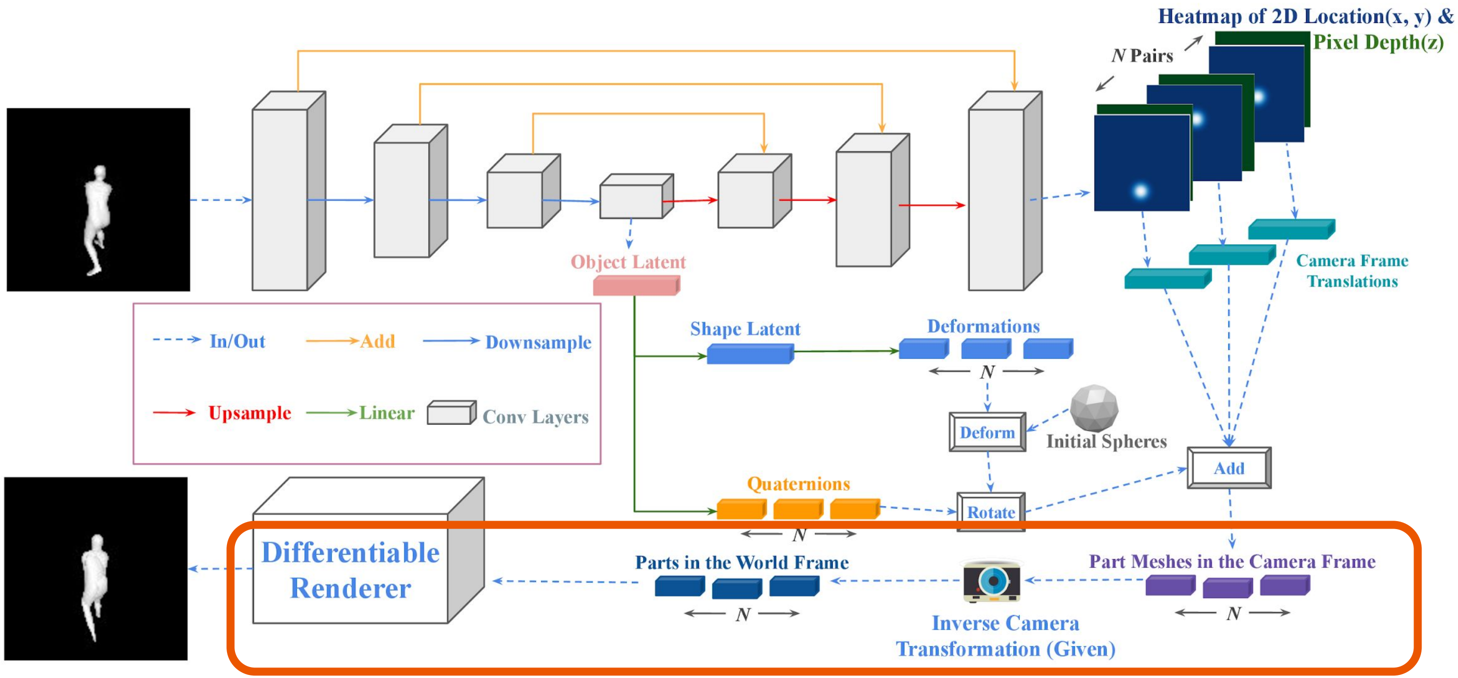


Cerberus: A Multi-headed Derenderer



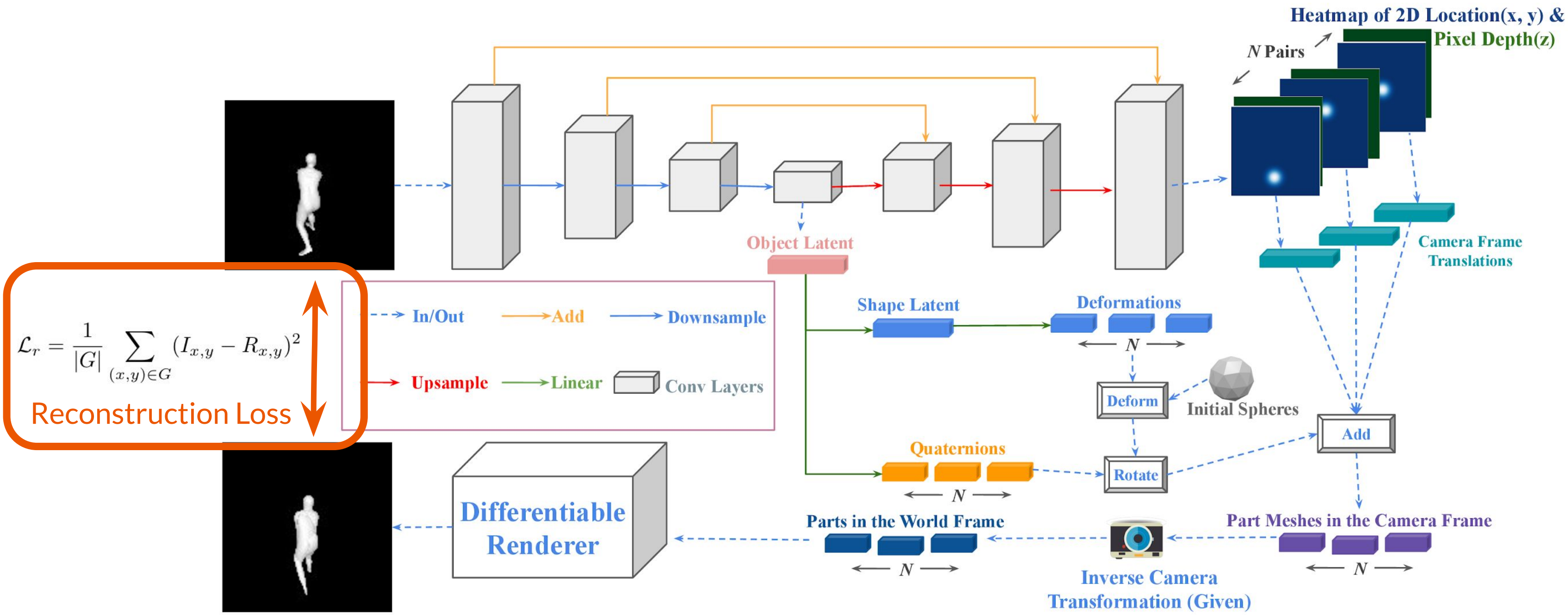
Mesh Generation: Latent -> Transformed Meshes

Cerberus: A Multi-headed Derenderer



Differentiable Mesh Rendering

Cerberus: A Multi-headed Derenderer



How to learn parts?

- With supervision (e.g. keypoints) -> Easy. Add a loss term.

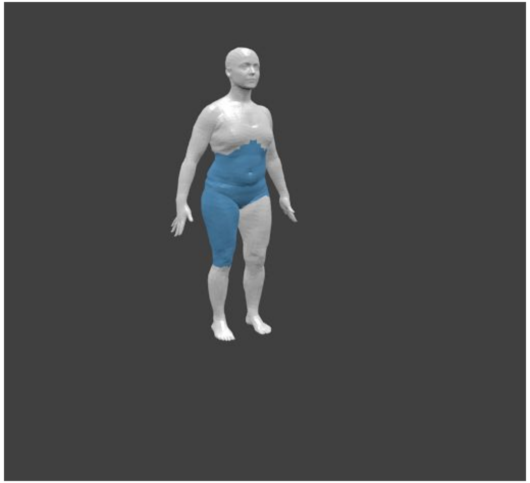
How to learn parts?

- With supervision (e.g. keypoints) -> Easy. Add a loss term.
- Can we learn parts without part annotations?

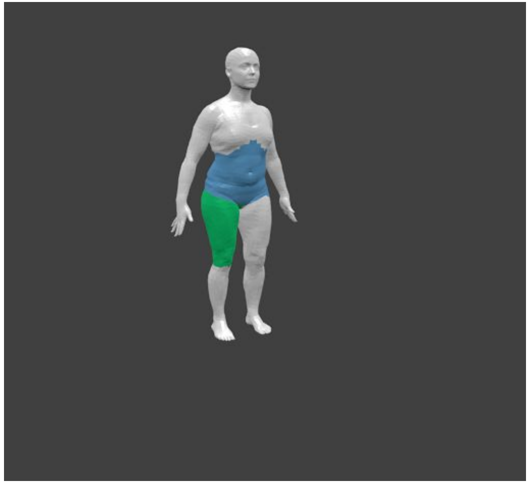
How to learn parts?

- With supervision (e.g. keypoints) -> Easy. Add a loss term.
- Can we learn parts without part annotations? **Yes.**
- Let's rethink the properties of parts.

How to learn parts?

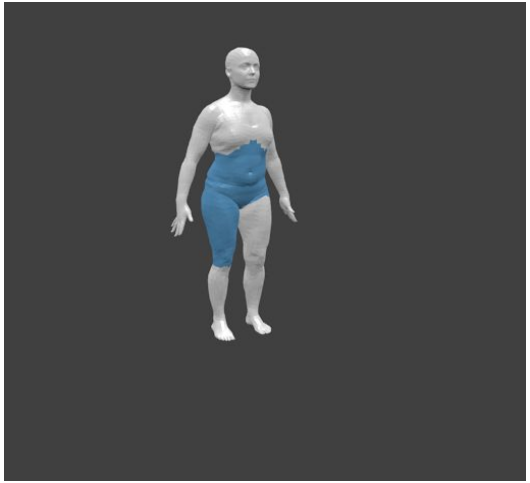


Part Split No.1



Part Split No.2

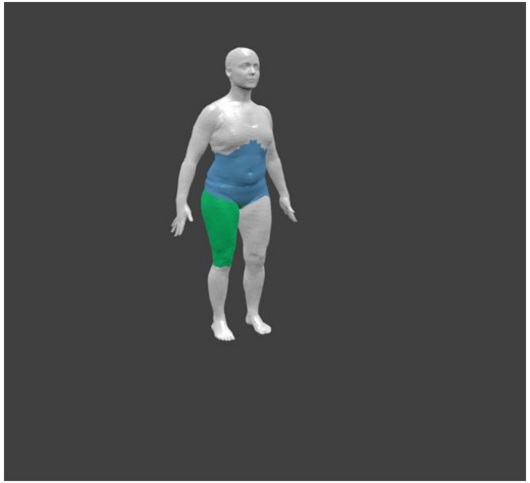
How to learn parts?



Part Split No.1

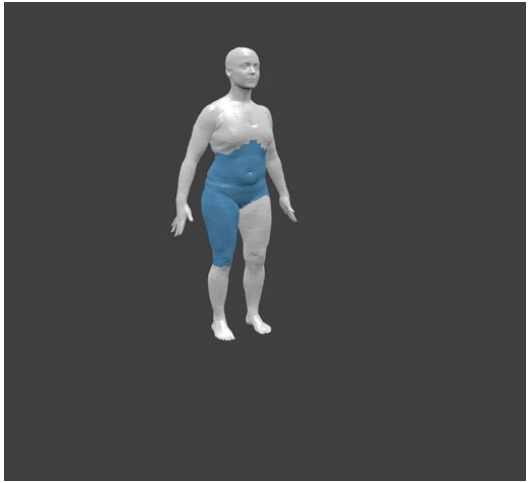
No.2 is preferred

Why?



Part Split No.2

How to learn parts?

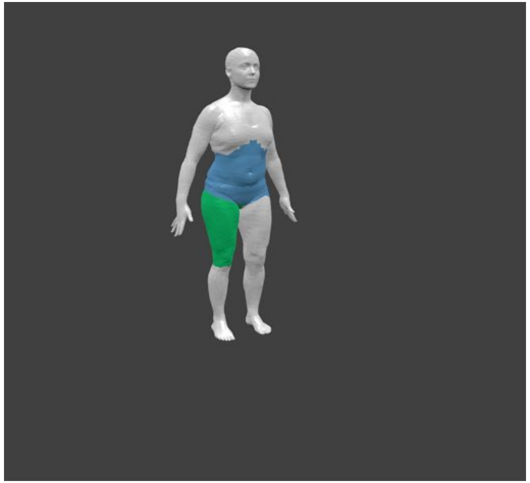


Part Split No.1

No.2 is preferred

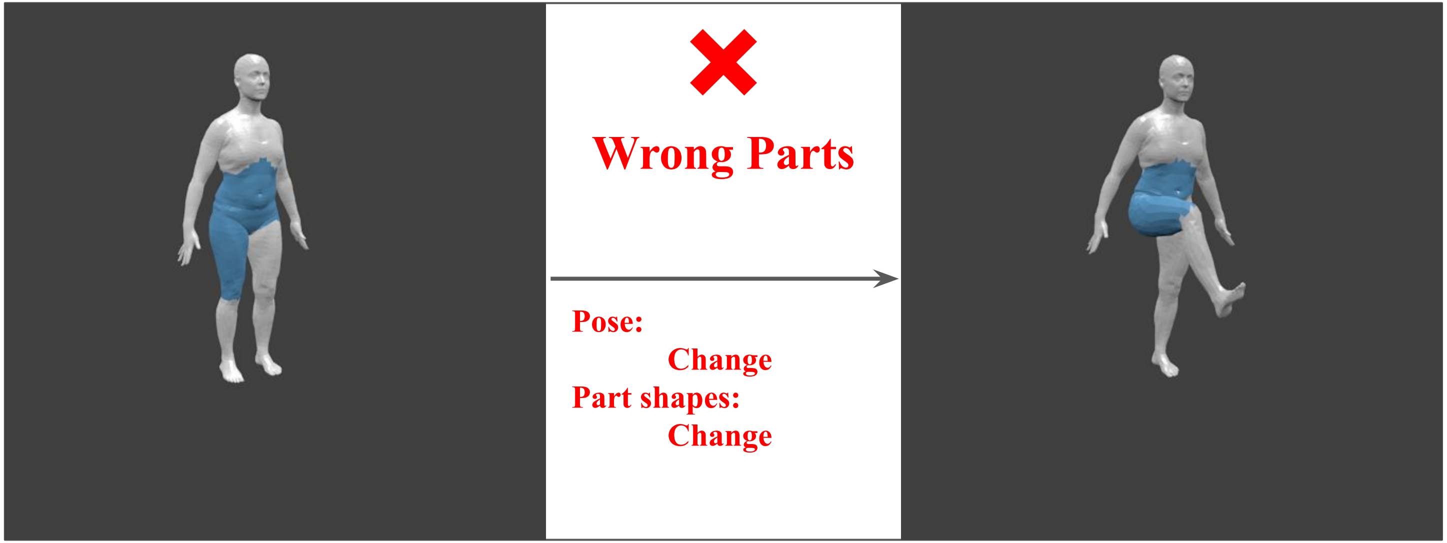
Why?

Pose Consistency



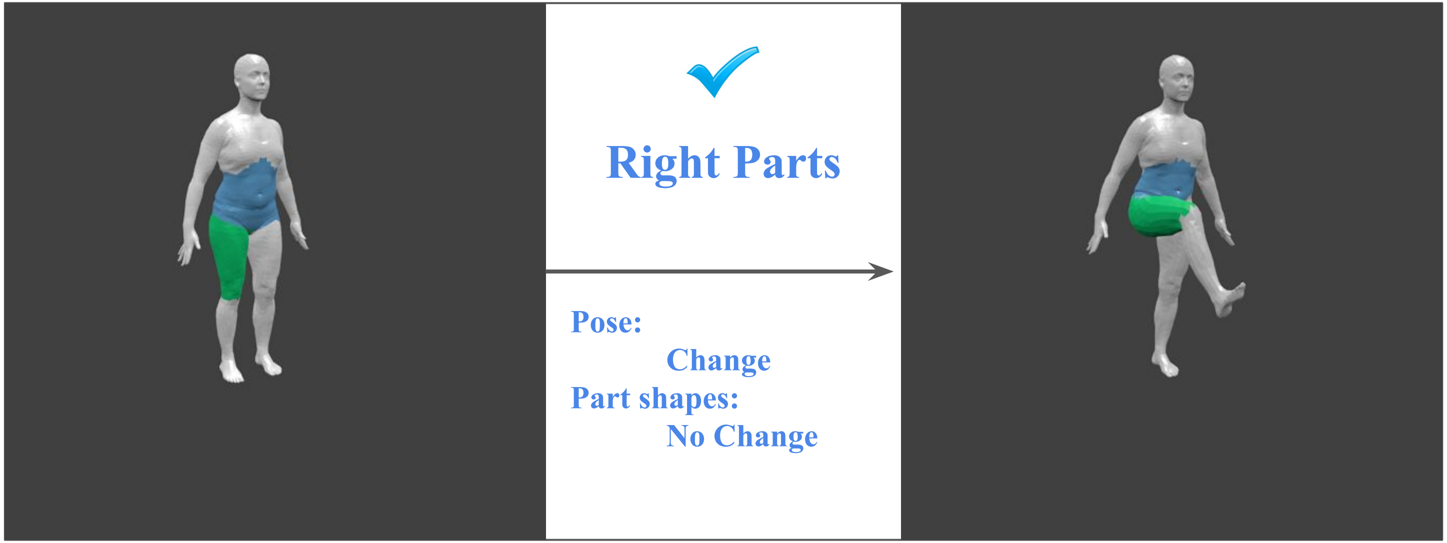
Part Split No.2

How to learn parts?



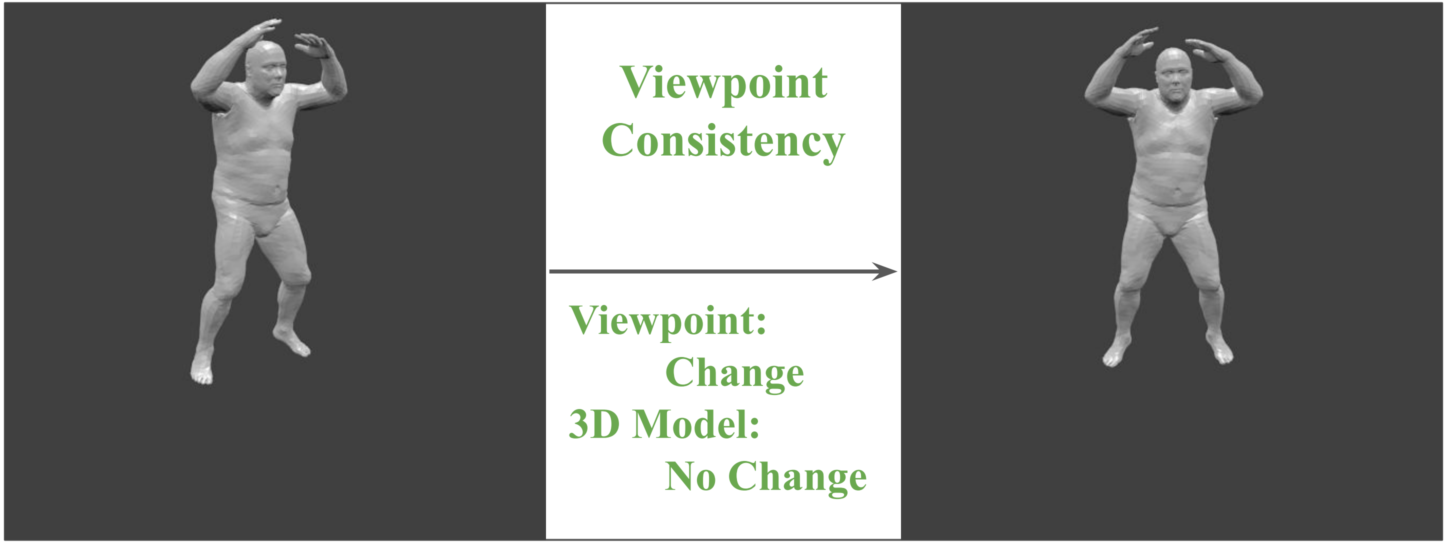
Part Split No.1

How to learn parts?



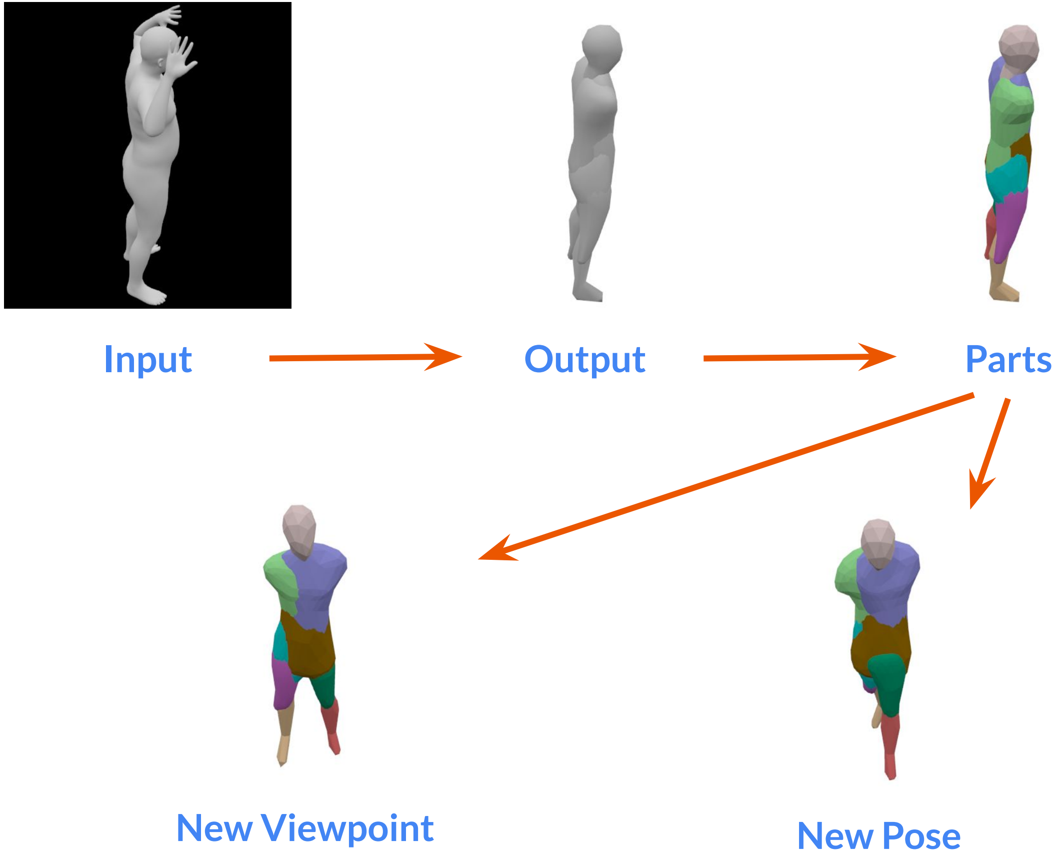
Part Split No.2

Another consistency



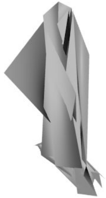
Viewpoint Consistency

Results



Results: Human Dataset*

Input



NMR

NMRs

NMRr

Ours

Our Parts

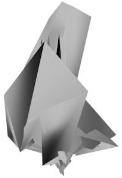
Our Turn

- NMR is Neural Mesh Renderer.
- NMRs is NMR with smooth loss.
- NMRr is NMR with our differentiable Renderer

*The dataset is credit to Angjoo's project(https://github.com/akanazawa/hmr) and SMPL model from MPII (http://smpl.is.tue.mpg.de/)

Results: Human Dataset*

Input



NMR

NMRs

NMRr

Ours

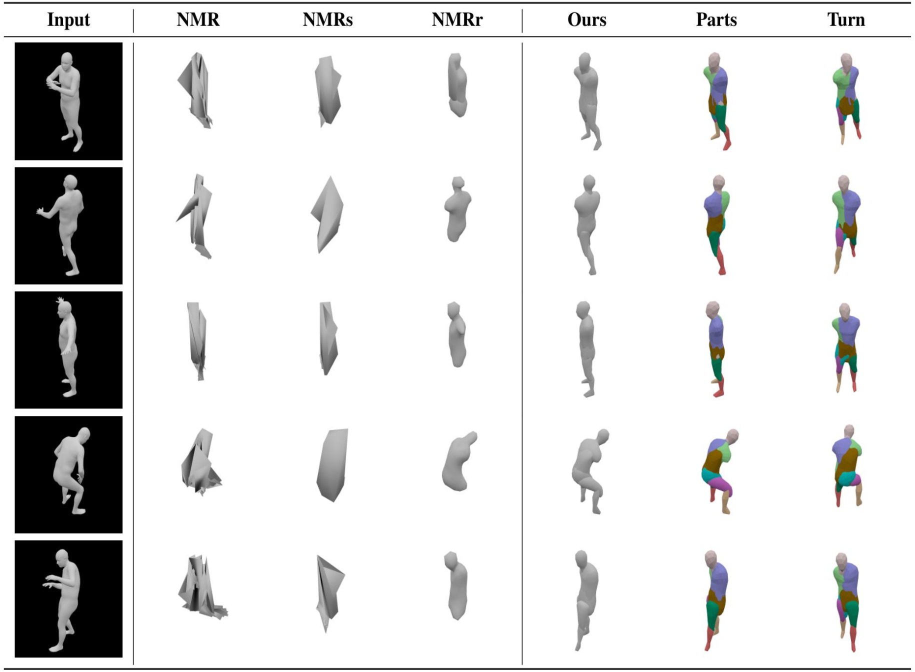
Our Parts

Our Turn

- NMR is Neural Mesh Renderer.
- NMRs is NMR with smooth loss.
- NMRr is NMR with our differentiable Renderer

*The dataset is credit to Angjoo's project(https://github.com/akanazawa/hmr) and SMPL model from MPII (http://smpl.is.tue.mpg.de/)

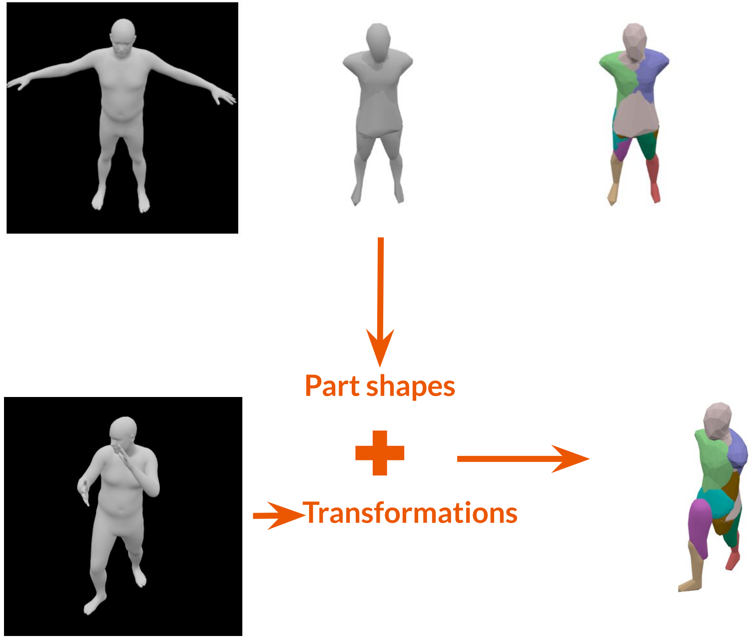
Results: Human Dataset



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

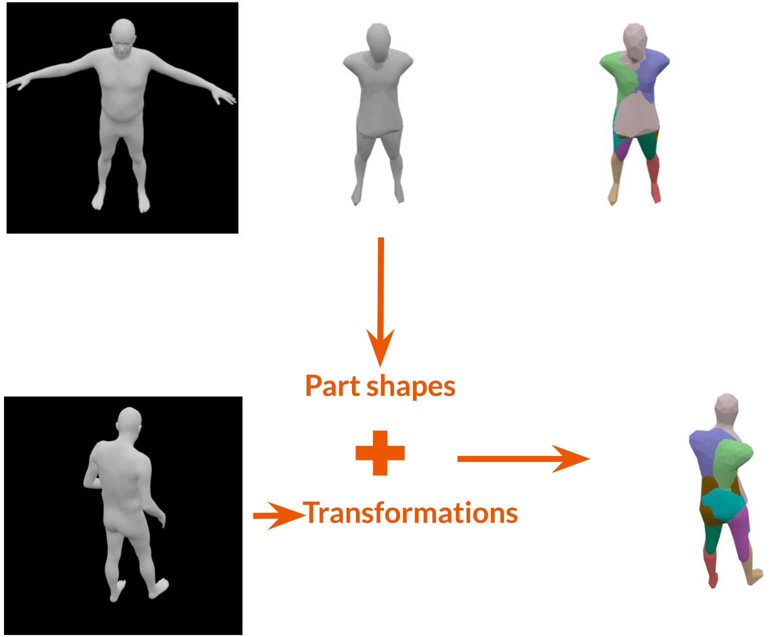
- Free Cerberus is Cerberus without pose consistency.
- Cerberus is better than baselines both quantitatively and qualitatively.

Results: Evaluate parts quantitatively



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

Results: Evaluate parts quantitatively



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

- The performance of Cerberus doesn't drop much on the hard test.
- Pose consistency can help learn better parts.

Results: Evaluate parts quantitatively

Input

Free Cerberus

Cerberus

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

- The performance of Cerberus doesn't drop much on the hard test.
- Pose consistency can help learn better parts.

Results: Animal Dataset* (Higher Shape Variance)

(a) Cougar

(b) Deer

(c) Tiger

(d) Hippo

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

- **Cerberus is consistently better than baseline methods.**

*The dataset is credit to SMAL from MPII.

Summary

- We present Cerberus, a 3D perception framework for articulated bodies.
- We present consistency constraints for learning parts without part supervision.
- Cerberus, trained with the proposed constraints, outperforms baselines on both standard and hard tests.

Thank you for listening!