Towards Object Reconstruction using Convolutional Mesh Regression

Apurba Sengupta
University of Pennsylvania
aseng@cis.upenn.edu

Abstract

3D reconstruction of objects from 2D images is an interesting research problem that has been of high interest to the Computer Vision and Deep Learning community for a long time. With the advent of deep neural architectures and high performance computing using GPUs, the push towards utilizing these concepts for achieving state-of-the-art perception results is fascinating. In this project, we explore few strategies for estimating 3D pose and shape from single images of objects like humans, cars, etc. In particular, we focus on mesh-based approaches to the above task, wherein an initial mesh structure is assumed that is consistent with the image evidence,. The mesh is then deformed using graph-based convolution operations, with image features encoded into the mesh structure, and the 3D location of the object is regressed out of the network.

Keywords *- 3D reconstruction, Deep Learning, mesh, graph-based convolution*

1 Introduction

The world we live in is a 3D one. We as human beings have exceptional intelligence when it comes to inferring 3D object shape from 2D photo of the same. Understanding the 3D property and features of objects is done very easily by the human vision system, but the same using Computer Vision is very difficult. Recent advances in Deep Learning techniques have tried to explore this problem of 3D reconstruction of object pose and shape from their 2D images. The idea is to use a deep neural architecture to achieve the required pose and shape of the object as regression outputs. Some results obtained by such a model is shown below in Figure 1. The first is the input image, the second the predicted 3D mesh structure and the third is the ground truth 3D mesh structure.

Figure 1: Input image, Predicted mesh, Ground truth mesh

In this project, we try to explore such techniques that use a general 3D mesh initialization of an object and embed the object's image features onto the vertices of this 3D mesh, before using graph convolutions to regress the 3D pose and shape of the object. The initial mesh is deformed along these 3D outputs to generate the 3D shape for the given object. This kind of an achievement finds excessive use in computer graphics, video games, AR/VR and self-driving car applications like 3D pose estimation of objects.

2 Related Work

The problem of 3D reconstruction of objects from 2D images has been explored by many researchers over the past few years. Initial approaches to solving this problem such as those by Guan *et al*[1] relied on annotating 2D keypoints and optimizing the parameters of the SCAPE[2] model for humans. This optimization procedure generated a mesh for the given images. Similar optimization based approaches were explored by Lassner *et al*[3] and Zanfir *et al*[4], who included semantic information from the images when optimizing the reconstructed mesh.

Most of the recent works for 3D reconstruction are based on the SMPL parametric model[5]. These models try different approaches to regress the pose and shape of the objects. Lassner *et al*[3] identify 91 landmarks on the human body surface and then regress the SMPL parameters using a random forest method. Pavlakos *et al*[6] also use keypoints combined with silhouettes of the image to regress the mesh pose and shape. Kanazawa *et al*[7] try to regress the SMPL parameters while minimizing the 2D keypoint reprojection loss in an adversarial setting to verify if the the final mesh is representative of an actual human pose and shape.

A more recent focus has been on using an initial mesh representation of the object and then deforming the mesh according to the learned pose and shape parameters. Kanazawa *et al*[8] try to utilize such a mesh representation for bird images, wherein the mesh is deformed towards the actual pose and shape, along with information about the object's texture to fully reconstruct a 3D representation of the bird. Wang *et al*[9] utilize an initial ellipsoid mesh structure for the objects, which is then progressively deformed using graph convolutions[10] and perceptual features obtained from the images. Kolotouros *et al*[11] employ a non-parametric approach in which they use the base SMPL template mesh for human representation and then regress the 3D location of the mesh vertices using graph convolutions and image features added to the graph vertices. They also allow for a coarse-to-fine adjustment of the regressed mesh pose and shape by allowing it to be passed through an SMPL regressor to generate the target pose and shape parameters.

Some of the results obtained by these methods are shown below (Figures 2 and 3). The regressed 3D pose and along with the ground truth image is shown. Different viewpoints of the predicted 3D shape are also shown, which suggest successful image completion. Figure 2 shows the output meshe generated with 2D-to-3D supervision (in light blue) and without any 2D-to-3D supervision (in light pink). Figure 3 shows the output non-parametric mesh (in light pink) and the SMPL mesh regressed from the former (in light blue).

Figure 2: Human 3D shape estimation by Kanazawa *et al*[7]

Figure 3: Human 3D shape estimation by Kolotouros *et al*[11]

3 Method

Our focus is on utilizing the non-parametric approach proposed by Kolotouros *et al*[11] for general object representation. We first briefly describe the specifics of the model and the loss functions then describe the kind of data we look at for the reconstruction task. The long term goal is to use such data representations

for extending the model from just human 3D pose and shape reconstruction to that for general objects that we see around us such as cars, chairs, tables, etc.

3.1 Model

The model proposed by Kolotouros *et al*[11] essentially consists of two main parts: an encoder network that is used to extract features from the image, and a decoder network that uses the extracted features along with the template mesh to generate the desired mesh. For the encoder network, a ResNet50[12] is used that has been pre-trained on ImageNet[13]. It essentially extracts a 2048-dimensional vector for a given input image. This 2048-dimensional vector is then attached to each vertex of the template mesh. The decoder consists of a series of graph convolutions with multiple residual connections and Group Normalization[14] operations. The graph convolutions are defined according to Kipf *et al*[10] as:

$$Y = \tilde{A}XW$$

Here, $X \in \mathbb{R}^{N \times k}$ is the input feature vector, $W \in \mathbb{R}^{k \times l}$ is the weight matrix and $\tilde{A} \in \mathbb{R}^{N \times N}$ is the row-normalized adjacency matrix of the graph. The regressed outputs of the graph convolutions are the predicted 3D vertices of the deformed mesh and three camera parameters. The general network architecture is shown in Figure 4 below. A more detailed architecture is described in Figure 5 (*next page*).

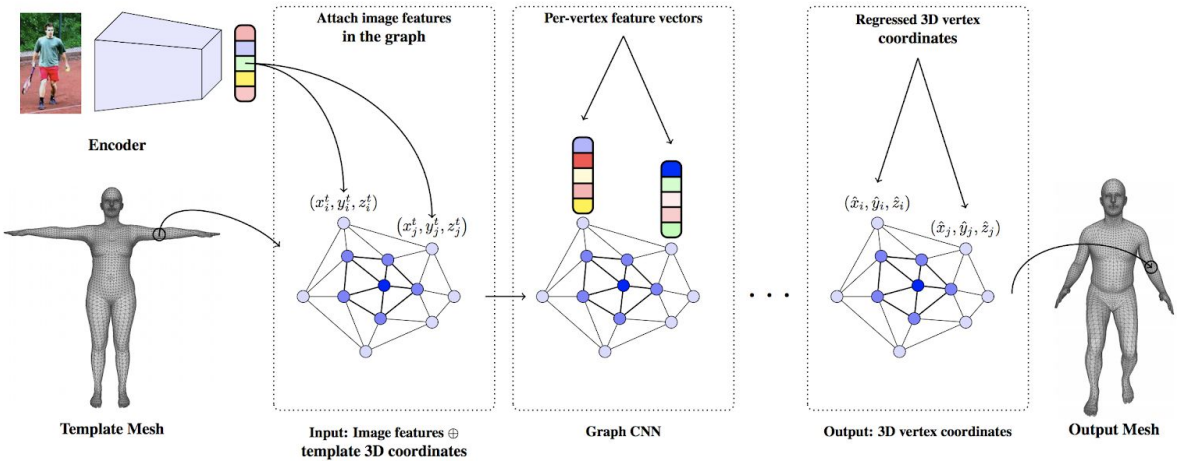


Figure 4: The encoder-decoder model used to regress the output 3D vertices of the mesh

3.2 Loss functions

The loss function essentially consists of three parts: (a) the mean-squared error[15] between the predicted 3D keypoints (which are obtained by applying a mesh upsampling procedure to the regressed 3D vertices) and the ground truth 3D keypoints, (b) the mean-squared error between the predicted 2D keypoints (which are obtained by projection of the 3D keypoints into the 2D space using the camera parameters) and

the ground truth 2D keypoints, and (c) the L_1 distance[16] between the predicted 3D vertices and the ground truth vertices. The total loss L_{total} therefore is a sum of these three losses:

$$L_{total} = \mathrm{MSE}_{3D} + \mathrm{MSE}_{2D} + L_1^{3D}$$

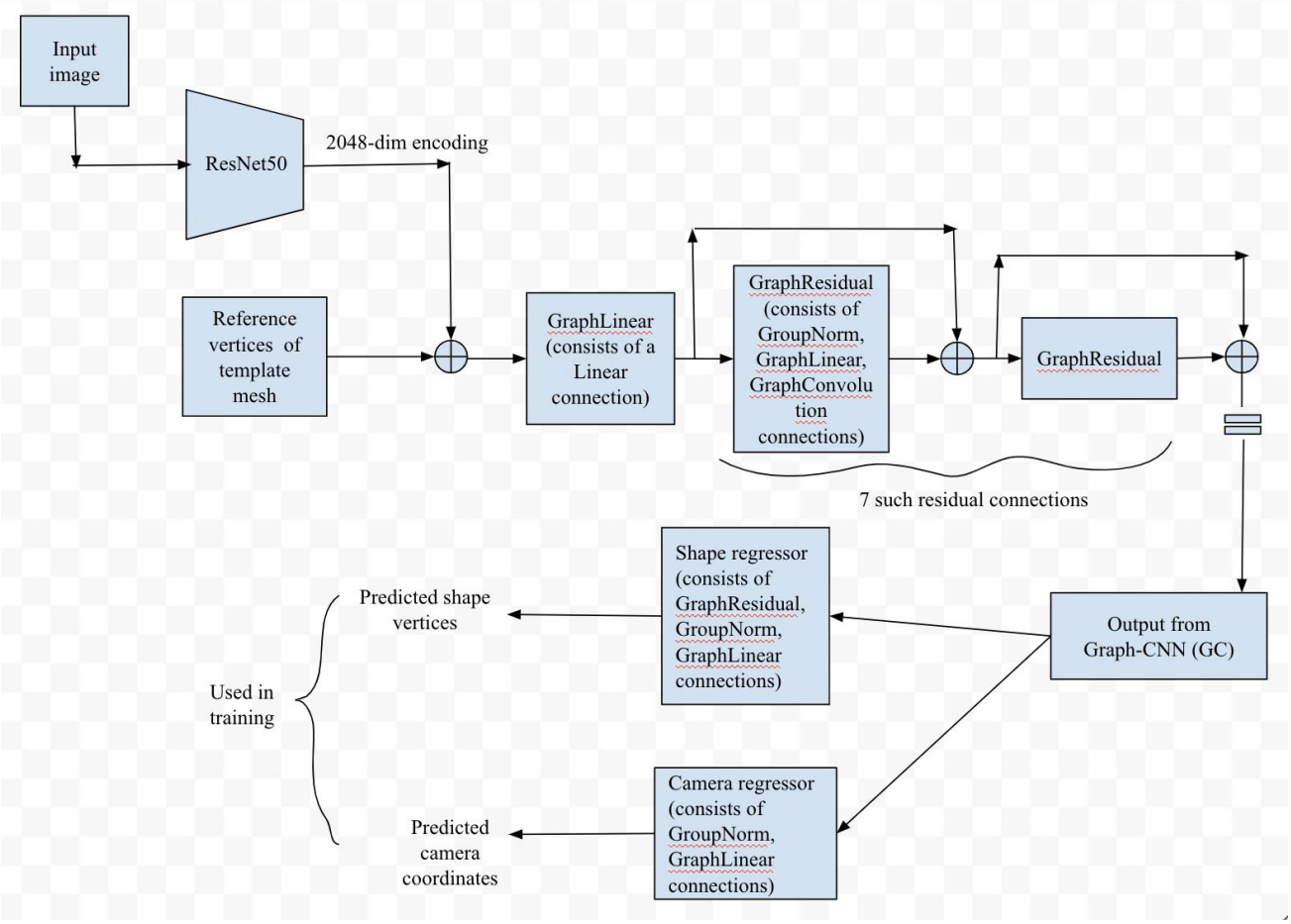


Figure 5: Detailed Block diagram of the graph convolution network to regress the 3D shape and pose of the object

(*Optional - Applicable to human 3D reconstruction*) The regressed mesh pose and shape can be fine-tuned by passing it as an input to a multi-layer perceptron classifier, which then outputs the SMPL pose (θ) and shape (β) parameters. The mesh is further deformed based upon these SMPL parameters. The mean-squared errors for the predicted SMPL parameters and the ground truth SMPL parameters are added to the loss objective above. So the total loss function L_{total} can be written as:

$$L_{total} = \mathrm{MSE}_{3D} + \mathrm{MSE}_{2D} + L_1^{3D} + \mathrm{MSE}_{\theta} + \mathrm{MSE}_{\beta}$$

An illustration for regressing the desired 3D pose and shape using the non-parametric 3D pose and shape obtained from the graph convolutions above is shown in Figure 6 below:

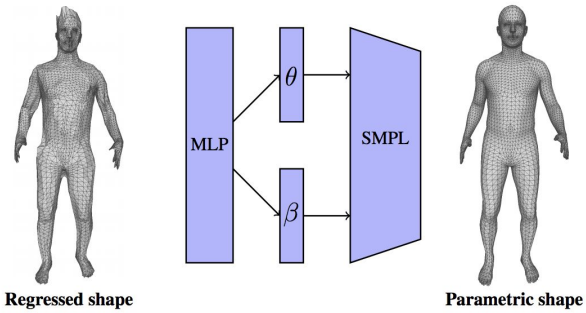


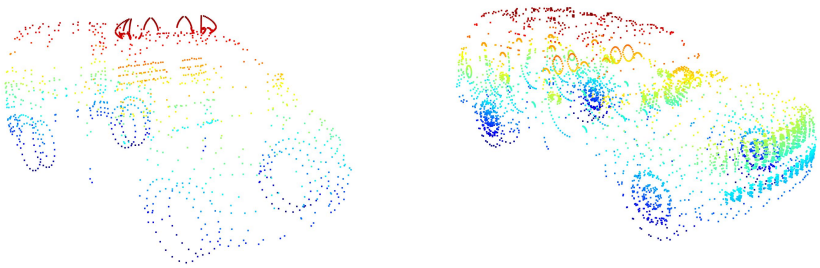
Figure 6: Regress SMPL parameters using the mesh structure predicted by graph convolutions

3.3 Data

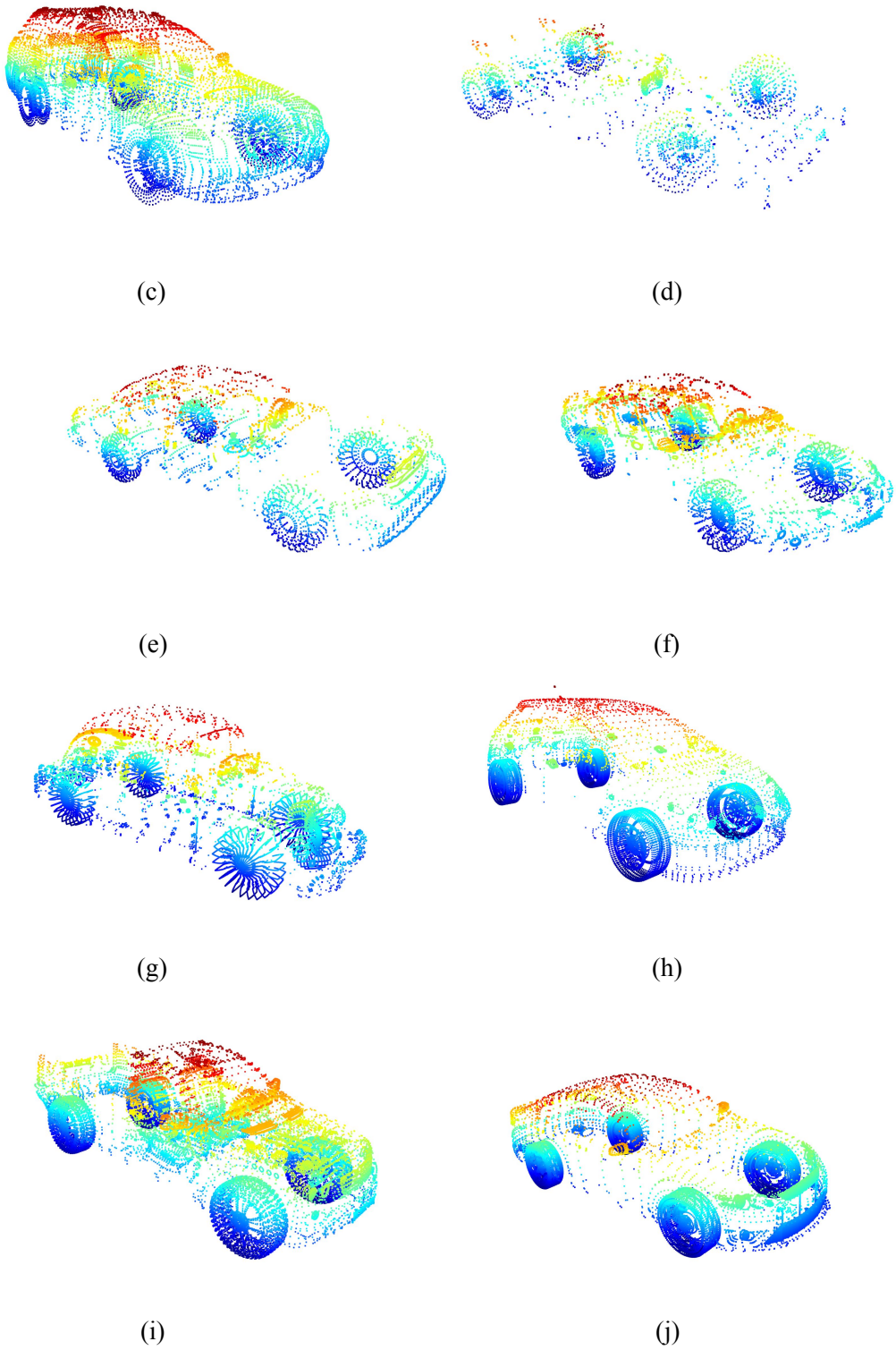
We utilize the PASCAL3D+[17] dataset for the purpose of 3D reconstruction of objects. This dataset contains images of objects in the wild and therefore pushes the need towards developing algorithms that are able to perform well on general objects rather than objects in a controlled environment like a laboratory. We mainly focus on car images in this project. Some of these images are shown below in Figure 7 below.

Figure 7: Cars in the wild - PASCAL3D+[17]

The cars dataset essentially consists of car images, their keypoint coordinates, camera features and CAD models. The CAD models contain 3D coordinates for 10 possible car configurations. The point clouds for the same are visualized below in Figure 8 with the help of Open3D[18] library.



(a) (b)



(c) (d)

(e) (f)

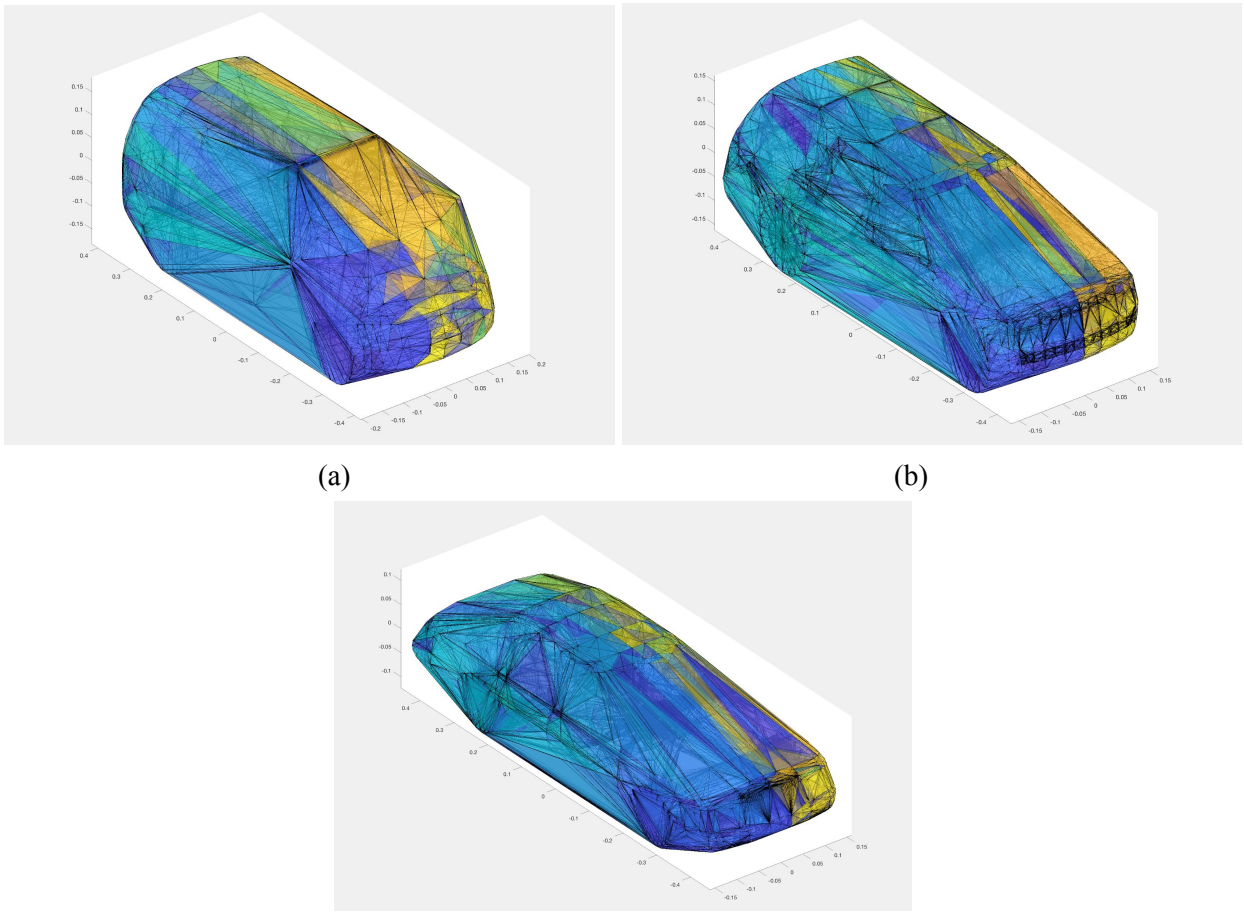
(g) (h)

(i) (j)

Figure 8: Point cloud images for the 10 given car configurations with (a) 6107 vertices, (b) 6341 vertices, (c) 17647 vertices, (d) 19016 vertices, (e) 28200 vertices, (f) 91566 vertices, (g) 111,504 vertices, (h) 123,239 vertices, (i) 160,278 vertices, and (j) 283,418 vertices

For each of the 10 configurations, the camera's azimuth and elevation are varied from 0 to 360° (with a skip of 5°) and -90° to 90° (with a skip of 2.5°) respectively. Thus, there are more than 5200 viewpoints generated for each configuration.

We also create meshes from the 3D coordinate data of the given configurations. We use Delaunay triangulation[19] for creating these meshes. Some of these meshes are visualized using MATLAB[20] in Figure 9 below.



(a) (b)

(c)

Figure 9: Mesh images for the car configurations with (a) 6107 vertices, (b) 6341 vertices, and (c) 28200 vertices

There are 12 keypoints identified for each car configuration. These are: (a) left front wheel, (b) left back wheel, (c) right front wheel, (d) right back wheel, (e) upper left windshield, (f) upper right windshield, (g) upper left rearwindow, (h) upper right rearwindow, (i) left front light, (j) right front light, (k) left back trunk, and (l) right back trunk. These keypoints (as red dots) are shown below in Figure 10 for one of the car configurations. The right front wheel and the right back wheel are not visible and hence not shown.

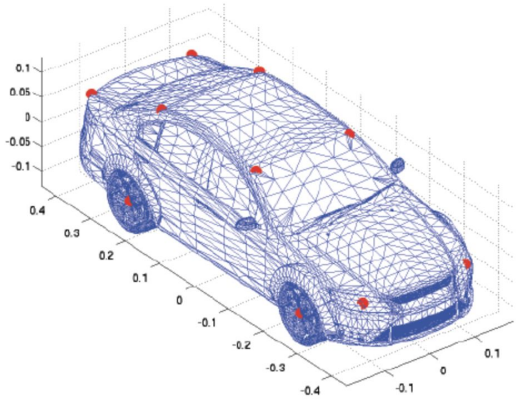


Figure 10: Car keypoints shown as red dots

Each of the 10 model configurations have the 3D coordinate information for these 12 keypoints. The images we have are for more than 2100 cars, with some of these keypoints visible in each of these cars. For each of these visible keypoints, we have the corresponding 2D coordinates. We also have the corresponding camera features like azimuth, elevation, distance, etc. We can infer the 3D coordinates of the image keypoints using the information about 3D coordinates from the configurations and the indices of the visible keypoints from the 2D coordinate information. These 3D and 2D coordinates essentially account as the ground truth 3D and 2D keypoint coordinates for the model described earlier in Section 3.1. The meshes generated using the Delaunay triangulation account as the template meshes for the same.

The necessary data are saved in NumPy arrays using `.npz` file format. The visible keypoints in the car images are available as `joint-ids.npz`, the 2D coordinates of these keypoints are available as `2d-keypoints.npz`, the camera coordinates corresponding to these keypoints are available as `camera-feats.npz`, the file indices for the cars are available as `file-ids.npz`, the 3D keypoint coordinates for the 10 car configurations are available as `3d-keypoints.npz`, the 3D keypoint coordinates of the cars for every configuration are available as `3d-keypoints-all-images.npz`, and the 3D coordinates of the 10 car configurations are available as `meshes.npz`. All these files are compressed into an archive called `req-fields.zip`.

4 Conclusions

We explored various approaches to 3D reconstruction of objects from 2D images. The main focus was however on utilizing a template mesh that is representative of the object we are trying to reconstruct, and then using a series of graph convolution operations to directly regress the desired shape and pose of the desired object. We first explored one such model for 3D pose and shape reconstruction of human beings and then tried to explore if such a model could be extended to general objects like cars, etc. In order to explore this problem, we looked at a dataset consisting of car images accompanied with their CAD models and annotations. From this data, we were able to extract 10 different car templates in their point cloud and mesh (Delaunay triangulation) representation, and the 3D location of 12 keypoints. We were

also able to obtain the 2D coordinates of these keypoints and the corresponding camera information of over 2100 cars from these images. Combining the 3D keypoint location information and the visible keypoints from the 2D coordinates, we get the 3D keypoint coordinates for the cars in the images.

5 Future Work

We can utilize the car images, the 3D and 2D keypoint coordinates, the corresponding camera coordinates and the template meshes as training data for the graph convolution model described earlier. We would need to define adjacency matrices and upsampling/downsampling matrices for the proper convolution operations. We can try to use image encoding techniques other than ResNet50 for extracting image features. Additionally, we can utilize other forms of information such as silhouettes, segmentation masks, etc. in order to extract more object features. We can also try to explore other datasets for cars that contain more than 12 keypoints and larger mesh structures, thus providing a richer set of information for keypoint and mesh representation. Finally, we can try to expand the model by incorporating objects other than cars and humans, such as chairs, tables, etc. so that we have a model for object understanding in general.

Acknowledgements

I would like to thank Prof. Kostas Daniilidis for giving me the opportunity to work with him on this project. I would also like to thank Nikos Kolotouros, PhD candidate in CIS, for his constant support in pointing towards resources and addressing my queries. A special thanks to the GRASP Laboratory and CIS department Professors, students and staff for arranging world-class seminars on the most current and relevant topics in Deep Learning that were directly or indirectly related to the topic of interest, and providing an integrated learning environment for intellectual and cultural exchange of ideas. Lastly, I would like to thank my parents and colleagues who have been a constant source of motivation and determination for me.

Once again, I would like to thank the above people. The completion of this project would not have been possible without their support.

References

[1] P. Guan, A. Weiss, A. Balan, M. Black, 2009, "*Estimating Human Shape and Pose from a Single Image*", URL - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5459300

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, 2005, "*SCAPE: Shape Completion and Animation of People*", URL - http://robotics.stanford.edu/~drago/Papers/shapecomp.pdf

[3] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. Black, P. Gehler, 2017, "*Unite the people: Closing the loop between 3D and 2D human representations*", URL - https://arxiv.org/pdf/1701.02468.pdf

[4] A. Zanfir, E. Marinoiu, C. Sminchisescu, 2018, "*Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes - The Importance of Multiple Scene Constraints*", URL - http://openaccess.thecvf.com/content_cvpr_2018/papers/Zanfir_Monocular_3D_Pose_CVPR_2018_paper.pdf

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. Black, 2015, "*SMPL: A Skinned Multi-Person Linear Model*", URL - http://files.is.tue.mpg.de/black/papers/SMPL2015_fixed.pdf

[6] G. Pavlakos, L. Zhu, X. Zhou, K. Daniilidis, 2018, "*Learning to Estimate 3D Human Pose and Shape from a Single Color Image*", URL - https://arxiv.org/pdf/1805.04092.pdf

[7] A. Kanazawa, M. Black, D. Jacobs, J. Malik, 2018, "*End-to-end Recovery of Human Shape and Pose*", URL - https://arxiv.org/pdf/1712.06584.pdf

[8] A. Kanazawa , S. Tulsiani, A. Efros, J. Malik, 2018, "*Learning Category-Specific Mesh Reconstruction from Image Collections*", URL - https://arxiv.org/pdf/1803.07549.pdf

[9] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, Y. Jiang, 2018, "*Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images*", URL - https://arxiv.org/pdf/1804.01654.pdf

[10] T. Kipf, M. Welling, 2017, "*Semi-Supervised Classification with Graph Convolutional Networks*", URL - https://arxiv.org/pdf/1609.02907.pdf

[11] N. Kolotouros, G. Pavlakos, K. Daniilidis, 2019, "*Convolutional Mesh Regression for Single-Image Human Shape Reconstruction*", URL - https://arxiv.org/pdf/1905.03244.pdf

[12] K. He, X. Zhang, S. Ren, J. Sun, 2015, "*Deep Residual Learning for Image Recognition*", URL - https://arxiv.org/pdf/1512.03385.pdf

[13] J. Deng, W. Dong, R. Socher, L. Li, K. Li and Li Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database*", URL - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5206848

[14] Y. Wu, K. He, 2018, "*Group Normalization*", URL - https://arxiv.org/pdf/1803.08494.pdf

[15] "*Mean squared error*", Wikipedia, URL - https://en.wikipedia.org/wiki/Mean_squared_error

[16] "*Taxicab geometry*", Wikipedia, URL - https://en.wikipedia.org/wiki/Taxicab_geometry

[17] Y. Xiang, R. Mottaghi, S. Savarese, 2014, "*Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild*", URL - http://cvgl.stanford.edu/papers/xiang_wacv14.pdf

[18] "*Open3D: A Modern Library for 3D Data Processing*", URL - http://www.open3d.org/docs/index.html

[19] "*Delaunay triangulation*", Wikipedia, URL - https://en.wikipedia.org/wiki/Delaunay_triangulation

[20] "*delaunayTriangulation - Delaunay triangulation in 2-D and 3-D*", MathWorks, URL - https://www.mathworks.com/help/matlab/ref/delaunaytriangulation.html