Unsupervised Single-View 3D Object Reconstruction

Abouzar Moradian

Project Report (First Update as of 11/3/2021)

- Summary of work to date: The topic I have chosen is 3D object reconstruction with the goal of shape completion from regular RGB images. The approach I picked was "signed distance function", or "SDF" as proposed by Park et al [1]. My original plan was to adopt a new version of SDF called "Signed Directional Distance Function" or "SDDF" recently proposed by Zobeidi [2], implement the algorithm as introduced in the latter paper, and then replace the training dataset which has been made from 3D model provided by ShapeNet dataset [3] with depth map made from regular RGB images. However, I ended up changing my approach (not the topic and the goal) because of the following reasons:

  - While I was writing the proposal, I had a good understanding of the general ideas of SDF and SDDF as well as the math behind them. I found the idea interesting and being aligned with my general topic which is shape completion. After the submission of my proposal, I spent three weeks going deeper into the details to start the implementation of the proposed algorithm as the first step of my project. However, I found some of the mathematical details, as well as some methodologies ambiguous. I tried to find the code to get clear on those ambiguities, but unfortunately, the code for SDDF is not publicly available. So, starting such a big project from scratch with lots of ambiguities is a risky task for a course project with limited time. So, my main concern was that I might not be able to finish it on time. (However, as I said, I spent a significant amount of time to go to the details and I had to refer to many other papers which was a good practice and led me to came across new ideas.)

  - After being done with SDDF, I decided to start with the implementation of the original algorithm, i.e., DeepSDF [1] whose code is available, and then go from there and incorporate new components. However, I found some challenges which made me believe that I might not be able to get my final goal if I take this path. The first reason was that there are several complicated preprocessing steps I would have had to take before I could feed the training data into the model. The training data were supposed to be random points in the predefined field containing the training 3D shapes as well as their corresponding signed distances. Since this kind of training dataset is not readily available, I needed to use 3D ShapeNet dataset [3] to make my own dataset through some complicated methods which was very time consuming making it probable to run out of time. In addition, these preprocessing steps would diverge from my main goal which is replacing the 3D shapes dataset as training data with regular RGB images. Therefore, it was not reasonable to spend the significant portion of my project

preparing the training dataset which were to be replaced with some simpler dataset as the main goal of my project.

- During my effort to go deeper into SDF algorithm, I came across some papers more relevant to my main goal, which are shape completion using multiple regular RGB images, which are simpler and more interesting. These methods are geometrically complicated as well but use some readily available datasets with relatively fewer preprocessing requirements, making it more attractive to me as an alternative to my previous approach.

- Analysis of Work: In my proposal, the goal for the first update were mentioned to be completing the implementation of SDDF algorithm, as well as coming up with some new ideas about how I can improve it in terms of using some simpler and more available training dataset. As I mentioned in the previous section, after working for a couple of weeks on the original paper, I realized that it is beyond a course project in terms of time and energy it demands. However, my achievements since I submit my proposal includes having a deeper understanding of the existing methods of 3D shape reconstruction, finding an alternative algorithm which uses regular RGB images as its training dataset with some really interesting methods to draw some geometrical meaning from multiple RGB images, and making some progress in implementation of the new algorithms which I will explain in the next section.

- *Plan for Completion: As I mentioned in the previous sections, the topic of my project remained unchanged, but my approach and algorithm changed based on my new findings. The paper I recently found, and I plan to do my project based on is an algorithm proposed by Ho et al [4] a couple of months ago which is an altered version of a method proposed by Wu et al [5]. What I really like about these papers is that they trained their 3D reconstruction models only based on one or two images of the objects, and both receive only one image in the inference phase as input making them more practical and versatile compared to those models which requires a 3D supervision in the training phase. Now I will briefly explain the two algorithms proposed by the two papers and then I will introduce my new goals.*

  - Wu et al [5], introduced a novel model which is able to learn 3D objects categories from raw single-view images without any supervision. This algorithm was designed for object with some symmetric structure, like human face, car, etc. This method's goal is to make a model which is able to learn a function mapping input image to four factors (d, a, w, l), which are "depth map", "albedo image", "global light direction", and a "viewpoint". Since there is a mathematical method to reconstruct an image given these four factors, if the model learns a function to derive these four elements from a single image, then

we will be able to reconstruct an image of the objects from any arbitrary viewpoint and lighting condition which means having a 3D model of that object. The model learns to output depth map and albedo of input image I from its canonical viewpoint. Therefore, since the assumption is the shape is symmetric in canonical view and given the fact that canonical albedo and depth map are independent of lighting and viewpoint, if we input the mirrored version of the original image, the model must give the same depth map and albedo. That way we can make our loss function based on two constrains, first, the reconstructed image from factors d, a, w, l, must approximate the input image, second, the reconstructed image of original input image and its mirror must be approximately the same. One more advantage of this method compared to the SDDF is that the former utilizes a pre-trained feature detector model like VGG-16 network to define perceptual loss function. That we will be able to incorporate some existing achievement into our model instead of starting from scratch.

- Ho et al [4] made some amendments to the original method proposed by Wu et al [5], to make it more versatile by removing the symmetry constrain. As a result, the model requires two images of each object in the training phase (as opposed to the original model which is trained on one single image) One advantage of this new version is that we can apply it to any shape without being required to be symmetric. In addition to this amendment, they introduced a new loss function called "albedo loss" to take care of the fine details of the shapes which otherwise will be lost as a result of over-smoothing.

- My new goal is implementing the algorithm proposed by Ho et al [4] as the first step of my project (which still is complicated and time consuming, but easier than the previous one I had chosen), and then trying to improve the model, or at least experiment the results of some changes, by manipulating the loss functions. I came up with the idea of manipulating the loss functions because in this paper, the focus and of course novelty of the algorithm are more on defining some appropriate loss functions instead of the architecture of the main neural network. It is my first time that I realized the importance of loss function in a neural network model. I found this approach interesting because the neural network itself is relatively simple and small, and the training data is extremely simple and readily available, but what makes the model strong is defining an appropriate loss function based on all our desired outcomes as well as existing geometrical functions and properties related to images and shapes. That's an important advantage over many other 3D reconstruction models which are based on a huge, complicated, and fancy architecture with a relying on a very sophisticated training datasets which are not available most of the time. So, my focus in this project is improving the model through loss function manipulation.

[1] Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 165-174).

[2] Zobeidi, E., & Atanasov, N. (2021). A Deep Signed Directional Distance Function for Object Shape Representation. *arXiv preprint arXiv:2107.11024*.

[3] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., ... & Yu, F. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

[4] Ho, L. N., Tran, A. T., Phung, Q., & Hoai, M. (2021). Toward Realistic Single-View 3D Object Reconstruction with Unsupervised Learning from Multiple Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 12600-12610).

[5] Wu, S., Rupprecht, C., & Vedaldi, A. (2020). Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1-10).