Deep Signed Directional Distance Function for Object Shape Representation Supervised by Still Images

Abouzar Moradian Tehrani
Abouzar.moradian@tamu.edu

- *Description of Problem*: Most of the promising shape completion algorithms including DeepSDF require 3D supervision (or depth maps, or calibrated images) which is not always readily available putting a serious limit on the scope of learning of the models. The goal is finding a way to train DeepSDF on regular still images instead of 3D shapes.

- *Importance of Problem: One of the most prominent limitation of any supervised model is the lack of enough training data. In 3D reconstruction tasks, it is almost impossible to make a through 3D dataset which covers all classes of 3D shapes and scenes in our environment. Even providing a set of calibrated images captured from different angels is not always feasible. So making a model, as strong as DeepSDF capable of learning from a variety of regular still images even when they have captured casually for some different purposes would be an astonishing progress.*

- *Proposal: DeepSDF [11] proposed an efficient and robust technique for shape completion tasks using "Signed Distance Function" [11]. However, it relies on 3D supervision which is not always available. Deep SDDF [12] is a variant of SDF with the capability of learning from depth maps instead of 3D shapes. They proposed the "signed directional distance function" which is determined by the distance of the observing point (depth camera) to the shape surface along the viewing direction. Therefore, all we need for the training dataset is depth maps captured from different angels. On the other hand,* Saxena et al [15] has shown that we can estimate 3D depth map from a single still image using supervised neural network. Therefore, putting these two steps together, it seems that we can train SDDF using still images instead of 3D shapes, or depth maps. Utilizing of still images seems easier on SDDF than other image-based methods such as view-synthesis models because in the latter type of models, the images must be calibrated and have to be correspondent to each other. (because of geometric correspondence the set of images are supposed to have). But in case of SDDF, since every depth image from a specific point of view contributes to the learning of the model independently of other views, we can easily pick some relevant still images belong to one class converting them to depth maps and feed to the model. Suppose we have an image of a car only from one view, but there are many other images online from the same type of car (maybe with different colors, and some slight differences), so we can put all of those images together and feed to the model under the same label, that way model would easily learn to reconstruct that car. The reason is that the model is completely flexible on the training data due to signed directional distance property.

(However, since I am not sure if it works as I am expecting, one alternative path would be improving SDDF in some different direction through applying "Curriculum" approach proposed by Duan et al [14] on SDDF making it more robust)

- List of goals:
  - First update: I expect to be able to implement the original SDDF algorithm (And hopefully DeepSDF as well) before my first update. That way I will have earned a good understanding of both algorithms and their details which would likely give me some better ideas compared to what I mentioned here.
  - Second update: At this point I expect I will be done with making necessary alterations on the original architecture to make it trainable with the sort of data I mentioned. I am hoping that manipulating the original architecture will make me comfortable with coming up with my own algorithm in the future.
  - Final Report: By the time I am supposed to submit my final report, I expect I will have reached some acceptable and presentable results on my experiment. Upon the completion of this project, hopefully, I will come up with some novel idea to pursue in the future.

Literature Review

Different approaches are usually used to model and represent 3D shapes and scenes. 3D reconstruction of an object or scene sometimes is performed in order to create a 3D model of a given object in the presence of that object. In this case we have a comprehensive spatial information about the object, i.e., information from different views, and the goal is to come up with a digital model of the object. But sometimes the goal is to find a method to reconstruct an object or scene in the absence of enough geometrical information about it using prior knowledge. For example, we are given a partial geometrical information about an object, and the goal it to complete the lost parts, on in other words, predict or reconstruct the lost part of the objects based on the given parts. The latter usually called shape completion. Since the main techniques are usually the same, I will explain each approach briefly, includes some seminal and some recent works on each and then will focus on the works have specifically been done on shape completion.

Mesh-based representation: in this approach, an object surfaces is reconstructed with a collection of faces (polygons) and vertices connected. As an example, for this approach, Sinha et al [1] proposed a method to construct 3D shape surface (rigid and non-rigid) using convolutional neural network. They expanded a "deep residual neural network" to generated some "geometry images" which represent each category of shapes. The model inputs 128×128 depth images for non-rigid shapes, and 128×128× 3 RGB images for rigid shapes and generates the geometry images as some

basis functions through which the varieties of shape surfaces, even unseen shapes can be reconstructed. [1] However, this approach faces several challenges including finding the optimum number of patches.

Another popular technique in mesh-based approach is mesh deformation technique in which the mesh is deformed using some various algorithm to reconstruct the target surface shape. Gao et al. for example, proposed a "sparse data driven mesh deformation" technique in which they sparsely picked basis deformation modes to reduce the dimensionality of their model and make the model more stable avoiding overfitting. The idea is that not every deformation mode contributes to the shape of objects. [2]

Litany et al used graph convolutional autoencoder to propose a mesh-based 3D shape completion method. Their model is able to learn latent spaces to complete the partial shapes. One of their contributions is doing the task of shape completion independently from learning the generative shapes models. They also introduced a novel graph convolutional autoencoder architecture which is very promising for any mesh-based 3D reconstruction technique. [3]

Bagautdinov et al is another example of mesh-based technique which used variational autoencoder to decompose the meaningful elements of geometry of human face which can both in recognizing human expression and generating new faces with a variety of geometric and expressive characteristics. [4] This method has shown a promising result, but the application is only limited to face, or any fixed template topology. So, it is hard to generalize it to cover a variety of shapes beyond those templates.

Point-based approach: in this approach the model directly works on a point cloud to represent the 3D shape. This approach is simpler than its counterparts firstly because it directly works on the raw data which can be easily earned from depth cameras or LiDARs. In addition, processing points are relatively straightforward, and the neural network to be used are simpler than other approaches. The main downsides of this approach, however, is that it cannot describe the topology of shapes, and it is a proper choice for making watertight surfaces.

Qi et al introduced their famous PointNet architecture as one of the most influential point-based method on 3D modeling. The novel thing about their work is that they suggested an approach to directly work on point clouds instead of transforming them into 3D voxel grids before feeding to the models. That way they preserved permutation invariance feature of point clouds. In addition, working directly on point clouds enabled them to make a simple and robust architecture, PointNet, which receives the point cloud as input, as classifies them to different classes of shapes or some components of objects or scenes. [5]

Although PointNet is very popular and influential, it only functions in classification and segmentation tasks. Yuan et al expanded point-based approach to cover shape completion tasks as well. They called their model "point completion network" or "PCN" which directly receives point cloud as input and learn to complete some partial point clouds. Their results is promising, but as

mentioned earlier, it is not able to describe the topology of the surfaces which is vital in many application.

Voxel-based approach: This approach might be a most normal and intuitive way of thinking of 3D objects. In this approach we work on the voxels which describes the volume of shapes in terms of 3D grids of values. Voxels are like pixels in 2D images. So, we can use 3D convolutional neural network to work on them. That way each voxel is either occupied by the volume or not which is called dense occupancy grid. One immediate problem that emerges from this method is its high computational cost as a result of cubical growth of computation demand.

One example of this kind of computationally expensive approach is what Wu et al proposed. They used a convolutional neural network to represent 3D shapes as "probability distribution of binary variables on a 3D voxel grid". [7] The problem with their method is that because it is computationally expensive, they had to limit their work on low resolution which sacrifices the fine details.

Tatarchenko et al proposed their Octree-based method to alleviate this problem. They presented an efficient deep convolutional decoder which avoid acting on regular voxel grids. In this method dense regular grids are replaced by octrees which save computational cost by preventing all of dense grids feeding to the next layer; only those regions with fine grained details needs those highly computational demands. [8]

Voxel-based Signed Distance Function: In this approach we still act on voxels grid, but instead of explicitly model the surface, we design a classifier which classifies each point into two classed based on their distance location relative to the surface. That way we implicitly give the surface as zero… SDF is a very popular technique to represent 3D objects and scenes.

Dai et al, for example, proposed a method for shape completion by utilizing TSDF. They combined volumetric deep neural network and 3D shape synthesis. Truncated sign distance field. That way a voxel outside the shape is positive which means empty, inside the shape negative which means unknown value and zero distance means on the surface. [9]

Zeng et al used truncated distance function (TDF) instead of TSDF, by removing the sign TDF is aimed to get improved in terms of concentration of larger value of voxel around the surfaces instead of between free space and occluded space which is an essential feature. [10]

The main problem with these conventional voxel-based SDF is that they are memory expensive due to use of discrete voxel which has let them to a low resolution shapes. DeepSDF [11] is a response to this issue. Park et al proposed a continues signed distance function which is more efficient that conventional discrete SDF because of representing a continuous implicit representation of surface of 3D shape. Magnitude a point is determined by its distance to the surface of the 3D shape. The points inside the shape are negative, outside positive, and on the surface zero. This efficiency enables this method to represent the shape with better resolution and details. [11] In addition, contrary to discrete SDF which only represent a single shape, DeepSDF

is capable of represent a class of shape by learn of latent space. This ability is super helpful in shape completion task because the model completes the shape based on the class the shape belongs to.

Although SDF outperforms compared to the conventional SDF, it is still relied on 3D shape to be supervised with. But the problem is that data for 3D supervision is not always available, and it is only limited to some 3D shape dataset such as ShapeNet dataset. Zobeidi et al has recently introduced a new version of SDF called SDDF or "signed directional distance function" which measure the distance in a given direction, when a scene is viewed from a given point of view, the distance is calculated between that reference point and the surface along that direction. That way, the training data can be earned from depth camera or LiDAR sensor which is much draw to earn compared to 3D voxel-based training data. In addition, unlike DeepSDF which requires some postprocessing such as surface extraction to visualize the result, SDDF directly deliver the rendering because the output of the model is distance in some given viewing directions which is a depth image or map. [12]

Since Zobeidi et al [12] has utilized a method called IGR introduced by Gropp et al [13], that might make sense to briefly discuss what they accomplished. Implicit geometric [13] regularization proposes a new method in computing neural representation shapes directly from point cloud. This model results in natural zero level through encouraging neural network to vanish on the input point cloud instead appalling to zero-loss solution.

Another interesting variant of DeepSDF has been proposed by Duan et al [14] called "Curriculum DeepSDF". They proposed a DeepSDF which gives different weights to hard and simple parts of training such that it learns to reconstruct course shapes first, and then gradually increase the accuracy by more penalizing the error to learn to reconstruct more complex details. The model outperforms DeepSDF in terms of reconstructing fine-grained details.

As mentioned earlier, Zobeidi et al [12] method has this advantage over DeepSDF that it does not rely on 3-D supervision which is not available always. Instead, the model inputs depth mapes which is easier to gather by depth camera. Another way to provide this depth images is by transforming regular image into depth image using the neural network model proposed by Saxena et al [15]. They applied a supervised model with a hug number of single still images and their corresponding ground-truth depthmaps as training data and managed to train the model such that inputs a still image and output a depth map.

References

[1] Sinha, A., Unmesh, A., Huang, Q., & Ramani, K. (2017). Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6040-6049).

[2] Gao, Lin, Yu-Kun Lai, Jie Yang, Zhang Ling-Xiao, Shihong Xia, and Leif Kobbelt. "Sparse data driven mesh deformation." *IEEE transactions on visualization and computer graphics* (2019).

[3] Litany, Or, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. "Deformable shape completion with graph convolutional autoencoders." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1886-1895. 2018.

[4] Bagautdinov, Timur, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. "Modeling facial geometry using compositional vaes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3877-3886. 2018.

[5] Qi, Charles R., Hao Su, Kaichun Mo, and Leonidas J. Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652-660. 2017.

[6] Yuan, Wentao, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. "Pcn: Point completion network." In *2018 International Conference on 3D Vision (3DV)*, pp. 728-737. IEEE, 2018.

[7] Wu, Zhirong, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. "3d shapenets: A deep representation for volumetric shapes." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912-1920. 2015.

[8] Tatarchenko, Maxim, Alexey Dosovitskiy, and Thomas Brox. "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088-2096. 2017.

[9] Dai, Angela, Charles Ruizhongtai Qi, and Matthias Nießner. "Shape completion using 3d-encoder-predictor cnns and shape synthesis." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5868-5877. 2017.

[10] Zeng, Andy, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. "3dmatch: Learning local geometric descriptors from rgb-d reconstructions." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1802-1811. 2017.

[11] Park, Jeong Joon, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. "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. 2019.

[12] Zobeidi, Ehsan, and Nikolay Atanasov. "A Deep Signed Directional Distance Function for Object Shape Representation." *arXiv preprint arXiv:2107.11024* (2021).

[13] Gropp, Amos, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. "Implicit geometric regularization for learning shapes." *arXiv preprint arXiv:2002.10099* (2020).

[14] Duan, Yueqi, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J. Guibas. "Curriculum deepsdf." In European Conference on Computer Vision, pp. 51-67. Springer, Cham, 2020.

[15] Saxena, Ashutosh, Sung H. Chung, and Andrew Y. Ng. "3-d depth reconstruction from a single still image." *International journal of computer vision* 76, no. 1 (2008): 53-69.

[16] Choy, Christopher B., Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction." In European conference on computer vision, pp. 628-644. Springer, Cham, 2016.

[17] Stutz, David, and Andreas Geiger. "Learning 3d shape completion from laser scan data with weak supervision." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1955-1964. 2018.

[18] Niemeyer, Michael, Lars Mescheder, Michael Oechsle, and Andreas Geiger. "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3504-3515. 2020.

[19] Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis." In European conference on computer vision, pp. 405-421. Springer, Cham, 2020.

[20] Curless, Brian, and Marc Levoy. "A volumetric method for building complex models from range images." In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 303-312. 1996.