

Reaction Report for "Deep Marching Cubes: Learning Explicit Surface Representations"

Nishant Raj

What I like about this paper?

One of the major missing components in surface shape reconstruction was the presence of an end-to-end surface prediction pipeline. Since Marching Cubes was not differentiable in nature, Deep Marching Cubes paper addresses that issue by creating a differentiable formulation of the loss function. It addresses this through four losses to capture different aspects: Point to Mesh Loss, Smoothness, Occupancy Loss and finally Curvature Loss. One of the good points about the architecture was that instead of using implicit surface representation, decoder predicts explicit surface representation. The way this helped was that this approach would only require sparse point cloud representation instead of dense ground truth of occupancy grid. One other modification that I liked was that instead of following PointNet++ they applied pooling on grouped points falling within a voxel in one set which helped them to leverage skip connections by maintaining the regular grid structure of the decoder. The authors have performed significant ablation studies like how different loss functions behave with respect to each other, validation in 2D data, displaying that the architecture is still able to predict the shapes well even in presence of moderate levels of noise in the dataset. These experiments show the robustness and validates their approach significantly. One of the final good things that they showed in their paper was how their method outperformed the baselines when it came to ground truth objects that were not closed in nature. This was due to the selection of occupancy loss as a component of the overall loss function.

What I do not like about this paper?

Though Deep Marching Cubes approach helps in the formulation of a differentiable approach of the marching cubes algorithm, I feel that it still has issues when it comes to scalability of the model. It converts entire point clouds into voxel grids and outputs an irregular mesh. This approach experiences cubic growth in computation with increase in resolution and hence it would only be scalable to few thousand points. A big reason for this heavy computation was presence of huge displacement ($N*N*N$) and occupancy tensors ($N*N*N*3$) placing a limitation on experimentations and resulting in failure to capture finer details. In the paper itself, we see that PSR-8 beats the performance of Deep Marching Cubes in completeness at higher resolutions. Due to these reasons, we see that the approach failed in cases like extremely thin surfaces or disconnected parts. Also, the experiments in the paper are done on the ShapeNet dataset. It would have been better if results from other datasets were used as well for comparison purposes.

Future Directions: To establish the robustness of the approach and model architecture, it would be interesting to see the comparative results on other different 3D datasets like KITTI and Online Products etc. and hence work needs to be done in this direction once the scalability issue is figured out. In terms of experiments, another reporting metric that could have been leveraged would have been volumetric IoU. Also, since we know that this approach does not scale well at higher resolutions, a good idea (as the authors have also suggested as well) would be to explore use of octree representations along with this architecture and explore other ways to leverage parallel computing. Inspiration for this could be taken from work like SSRNet. We also see that there were cases like extremely thin surfaces or disconnected parts where the architecture did not work well. A possible area of exploration to resolve this issue could be modifications to the smoothness and curvature loss functions defined in the paper.