# Lab 1: Deform Source Mesh to Target Mesh CV, week 3 presented by Georgia Gkioxari


## Student: Mory Moussou Koulibaly Traore

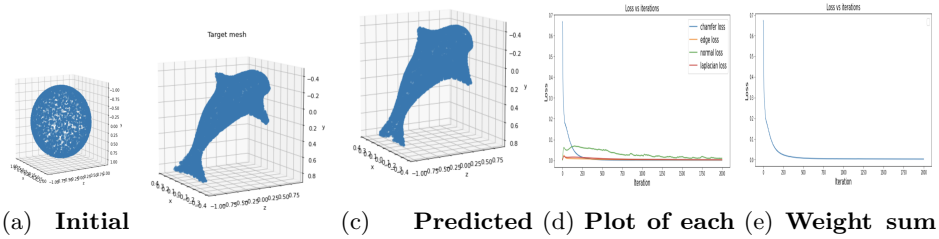
### AMMI/AIMS April 2021


## 1 Introduction

The purpose of this lab is to to familiarize ourselves with 3D data structures and differentiable 3D operators. We we learn how to deform an initial generic shape (e.g. sphere) to a target mesh .

## 2 Interpretation of our first study of the notebook

By looking our first notebook sended, we notice that we had some results here. And the first was after starting from a sphere mesh(sphere) showed in (**fig 1a**),we learned the offset to each vertex in the mesh such that the target 3D mesh shape (dolphin) in (**fig 1b**) is closer to the predicted mesh shown in (**fig 1c**) at each optimization step.And we have also the plot of each loss in (**fig 1d**) and figure of the sum of their weights in (**fig 1e**). We observe here all the curves decrease during the iterations.




(a) **Initial sphere** (b) **Target mesh sphere** (c) **Predicted sphere** (d) **Plot of each loss** (e) **Weight sum of the loss**

Figure 1: **Deformation of a source mesh and plot of the losses**

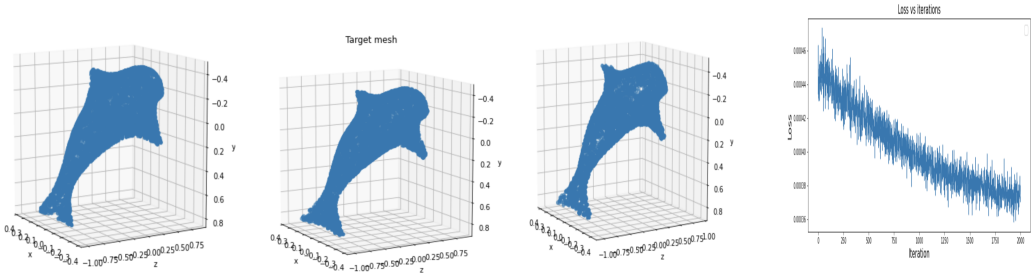
## 3 Part A: 3D Operators

### 3.1 Description of the purpose of each loss

- **Chamfer loss:** chamfer distance is the the sum of L2 distance to each point's nearest neighbour in the other set. It penalize mismatched positions between two points clouds. But it is not sufficient to produce nice 3D mesh.
- **Normal loss:** It is the observed surface normal from ground truth. And this loss is optimized if the normal of locally fitted target place is forcing to be consistent with the observation, which works practically well in our experiment.
- **Laplacian loss:** it help to prevent the vertices from moving to freely which potentially avoid mesh self-intersection.The laplaician term serves as a local detail preserving operator, that encourages neighboring vertices to have the same movement.
- **edge loss:** this one help to penalize flying vertice, which usually cause long edge, which add an edge length regularization.

### 3.2 set all loss weights to 0.0 except for the chamfer weight

When do this, we notice that the predicted mesh with regularizer shown in (**fig 2a**) is smoother and closer to the target shown in (**fig 2b**) than the predicted mesh without regularizer shown in (**fig 2c**). And When we plotted the weighted loss with respect to iterations shown at (**fig 2d**). It shows that the model with regularizer converges faster.

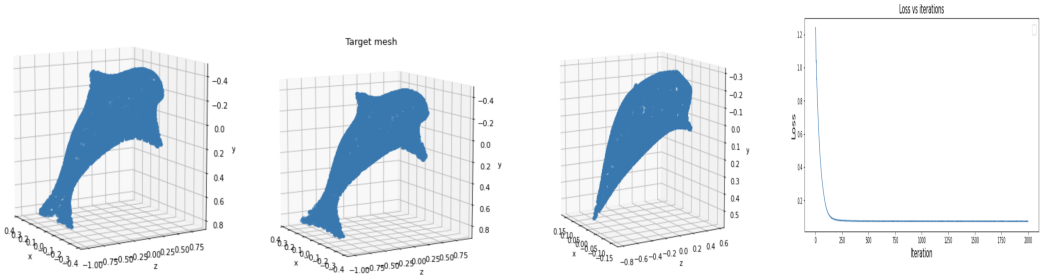



(a) **Initial sphere** (b) **Target mesh** (c) **Predicted sphere** (d) **Plot of weighted loss**

Figure 2

### 3.3 set w_edge to 100.0?

Here when we set the set w_edge to 100.0, We notice that predicted mesh shown in (**fig 3c**) miss some important part of the target (**fig 3b**). So it is less closer to the target than the predicted mesh shown in (**fig 3a**) with setting of w_edge to 1.0. But the weighted loss (**fig 3d**) has a more fast convergence.




(a) **Initial sphere** (b) **Target mesh** (c) **Predicted sphere** (d) **Plot of weighted loss**

Figure 3

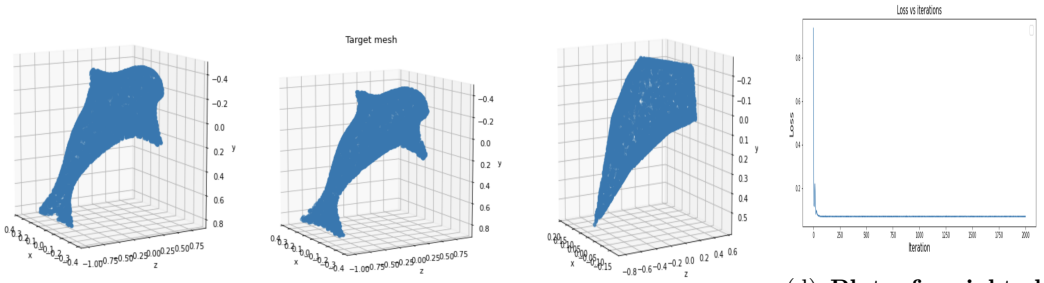
## 4 Part B: 3D Data Structures

### 4.1 Number vertices and how many faces of src_mesh

**src_mesh** has 2562 vertices and 5120 faces at level 4.

### 4.2 set the level of the sphere to be 1

Here, we setted the ico-sphere (level=1,device ) for src_mesh. And At level 1 , **src_mesh** has 42 vertices and 80 faces. And we observed that the predicted mesh at level 4 shown at (**fig 4a**) is better than the predicted mesh at level 1 at (**fig 4c**). At level 1, we have less verts and faces and the set from which we sampled point clouds is smaller and not enough to capture all parts of the target mesh shown at (**fig 4b**). But the weighted loss (**fig 3d**) converge faster than the other. And this starts after few iterations (around 49 iterations).




(a) **Initial sphere** (b) **Target mesh** (c) **Predicted sphere** (d) **Plot of weighted loss**

Figure 4

## conclusion

his tutorial allowed to learn how to load a mesh from an .obj file,how to use the PyTorch3D Meshes datastructure, how to use 4 different PyTorch3D mesh loss functions and how to set up an optimization loop