Using Neural Networks to Determine the Maximum Simplification of a 3D Mesh

Lauren Gardner | COS IW 03 | Dr. Xiaoyan Li

Motivation & Goal

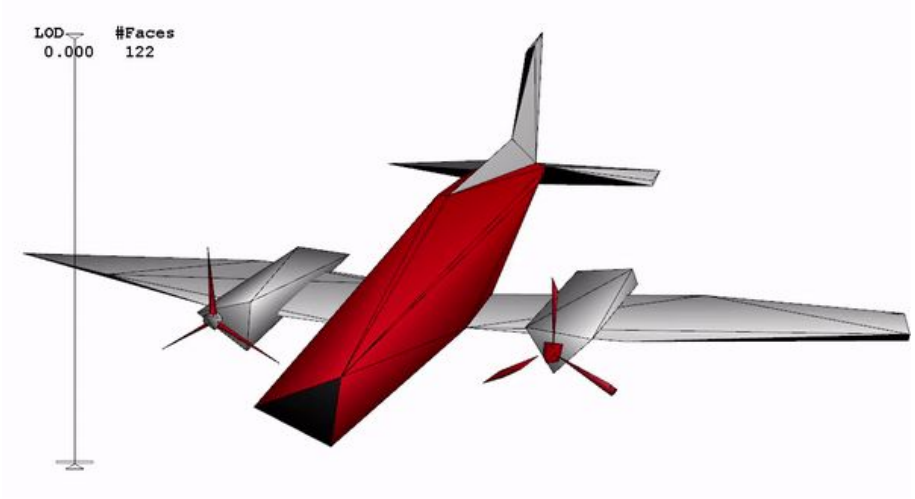
- Capabilities of computer graphics have greatly improved → detail of 3D meshes is increasing
    - **Increased detail = Increased latency**
- **Dynamic rendering** is a solution
    - Algorithms and methods that make it easier for scenes to be rendered quickly
- I used a convoluted neural network to identify the ideal base mesh needed to implement a **progressive mesh** [defined on next slide].
- **Goal:** Answer the question, what is the most simplified 3D object can be while still being recognizable – and does this depend on the object being represented?

Related Work - Progressive Mesh

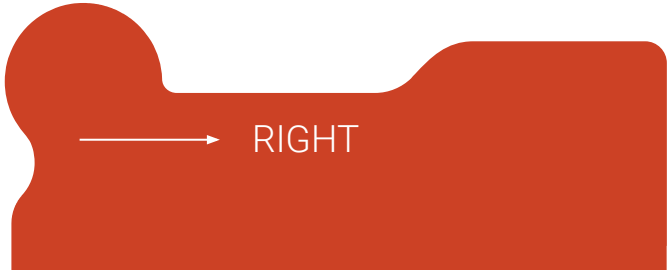
Progressive mesh → first introduced by Hugues Hoppe

Contains two objects: a **base mesh** and a list of **vertex splits**

- **Base mesh** – a mesh with the minimum amount of detail needed to represent an object
- **Vertex splits** - instructions sent to client that split the current mesh until the final level of detail (LOD) is obtained



Example of progressive mesh - Hugues Hoppe (1996)



Related Work - 3D Object Classification Models

There are multiple methods to classify 3D objects

1. Using a point cloud and deep neural network
2. Analyzing the curvatures of a 3D object
3. Taking snapshots of a 3D model at different angles and applying a standard 2D image classification

I used the third method to create my own 3D object classification model and determine the minimal amount of “detail” needed for this model to identify a 3D object

Methods

Main Goal: Create a 3D object classifier that can determine if a mesh is recognizable at certain stages of simplification

- **Classification concept:** classify 3D object using 2D photos
  - Visually based classification method
  - Most accurate when judging the likeness of one object to another in terms of visuals
- **Machine Learning Model:** Convolutional Neural Network (CNN) - 3 convolution layers
  - Commonly used for image classification because of high dimensionality

ModelNet10 → ModelNet2D

desk_0055 from all 5 views

Problem: ModelNet10 contains 3D mesh files (.off) – need a 2D equivalent

Solution: ModelNet2D - Collection of 2D images using meshes from ModelNet10

- **Scene**:
  - 5 camera locations: (0, -3, 0), (2, -2, -2), (2, -2, 2), (-2, -2, 2), and (-2, -2, -2)
- **File system**:
  - 5 destination folders for each object and the associated view angle.

Implementation

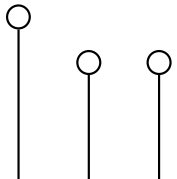
- **Libraries:**
  - ‘io’ → import dataset, save pngs to Google Drive
  - ‘pandas’ → manipulate and work with dataset
  - ‘open3d’ → import .off file & simplify mesh
  - ‘matplotlib’ → display meshes
- 1 Training set, 4 Testing sets - each with a decreasing percentage of triangles
  - Testing sets: ‘Test_0’, ‘Test_30’, ‘Test_50’, ‘Test_70’ – (simplified using open3d built-in library)
  - Wrote code to process 3D models into image representation described & to reduce # of faces for each test set
- 5 models - 10 classifications per model

RIGHT

Trained Models

CNN

- 20% Validation split
- Trained for classification at each angle (GPU constraint)
  - Ex: Train_0 → training set for view '0' with classes = 10 original objects
- 5 angles = 5 trained models



Train_0

Loss: 0.1079
Accuracy: **0.9652**

Val Loss: 0.6597
Val Accuracy: **0.8531**

Train_1

Pending...

Pending...

Train_2

Pending...

Pending...

Train_3

Pending...

Pending...

Train_4

Pending...

Pending...

Test_0: 0% simplification
Test_30: 30% simplification
Test_50: 50% simplification
Test_70: 70% simplification

View 0 Model

Test_0

Overall accuracy: **91.47**

```
Bathtub_0 accuracy: 79.59812658614126
Bed_0 accuracy: 90.89101824240807
Chair_0 accuracy: 83.52949115736732
Desk_0 accuracy: 89.72565134366353
Dresser_0 accuracy: 94.09871578701143
Monitor_0 accuracy: 81.9031739029391
Night_0 accuracy: 95.68406957568544
Sofa_0 accuracy: 80.0092987716198
Table_0 accuracy: 98.24912390682135
Toilet_0 accuracy: 96.54649071461331
91.47097063137264
```

Test_30

Overall accuracy: **91.36**

```
Bathtub_0 accuracy: 79.82169117393165
Bed_0 accuracy: 90.1287490167679
Chair_0 accuracy: 81.5864618996094
Desk_0 accuracy: 89.0038008641715
Dresser_0 accuracy: 94.83009496840035
Monitor_0 accuracy: 82.40602042140632
Night_0 accuracy: 95.77847076485855
Sofa_0 accuracy: 80.04652801901102
Table_0 accuracy: 98.34669095076872
Toilet_0 accuracy: 96.81829127062738
91.35745263438884
```

Test_50

Overall accuracy: **91.44**

```
Bathtub_0 accuracy: 79.79669863807744
Bed_0 accuracy: 90.27025768390068
Chair_0 accuracy: 82.57631746859386
Desk_0 accuracy: 89.42131996154785
Dresser_0 accuracy: 94.94751364719576
Monitor_0 accuracy: 82.1905376068477
Night_0 accuracy: 95.39018082197266
Sofa_0 accuracy: 80.33624151721597
Table_0 accuracy: 98.4402892629752
Toilet_0 accuracy: 96.95389534519838
91.43606973433397
```

Test_70

Overall accuracy: **91.05**

```
Bathtub_0 accuracy: 79.61213465394646
Bed_0 accuracy: 89.71349843419515
Chair_0 accuracy: 82.9651422541717
Desk_0 accuracy: 88.8204924385957
Dresser_0 accuracy: 94.86096007067982
Monitor_0 accuracy: 81.59998516584265
Night_0 accuracy: 95.67098009465921
Sofa_0 accuracy: 80.71919241920114
Table_0 accuracy: 97.6556200782458
Toilet_0 accuracy: 96.87264725170304
91.0476736716575
```

UP

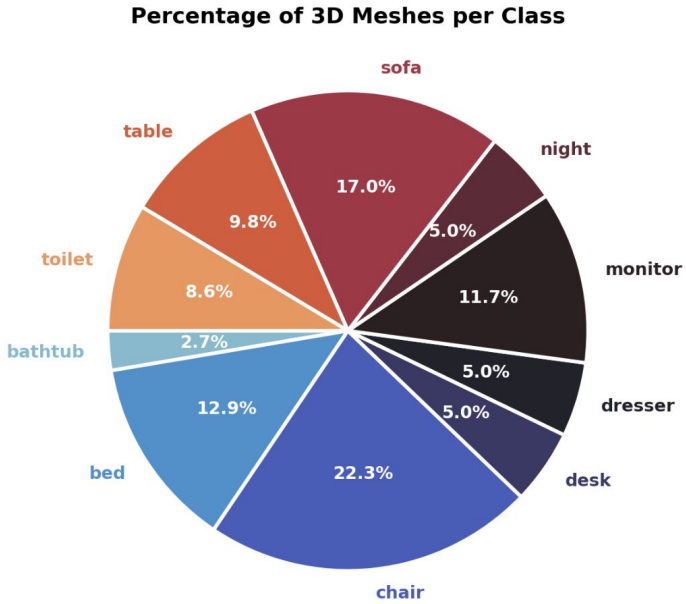
Train 0 - Break Down

Highest Accuracy: Table ~%98
Lowest Accuracy: **Bathtub** **~%79**
Hypothesis:

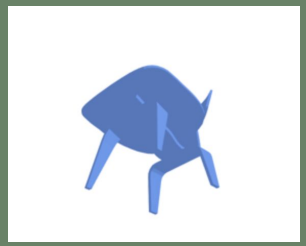
- Percentage of bathtubs much **lower** compared to other objects

Accuracy between simplification were not significant

- The simplification method used was built into the open3d library
- Wide variety of # of faces per mesh – most remained unaffected
  - Simple objects: table, desk, bed, sofa
  - Detail is lowered, basic shape remains → classification remains same



Views 1, 2, 3, 4
Results: Pending ...




3M
High Density

Conclusion & Future Work

The test results from before show that:

- It is possible to simplify a 3d mesh by a large degree (up to 70%) and not losing identification capabilities

Future Work

- Same process but with ModelNet40 (more objects)
- Increase the number of classification angles
- Edit simplification method

Thank You