

Deep Meshes

Luiz Velho
IMPA

Based on Slides by J. Johnson and E. Rodola'

Outline

- Mesh Representation
- Pixel2Mesh
- Mesh R-CNN

3D Mesh Representation

3D Shape Representations: Triangle Mesh

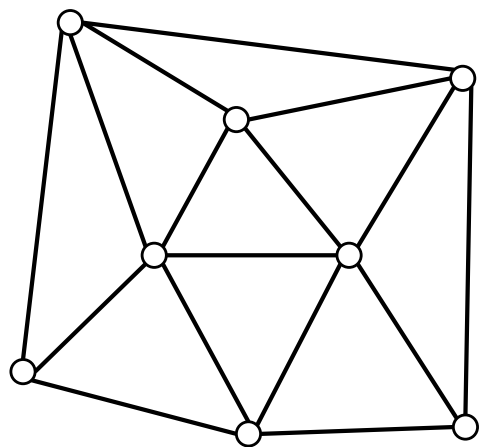
Represent a 3D shape as a set of triangles

Vertices: Set of V points in 3D space

Faces: Set of triangles over the vertices

(+) Standard representation for graphics

(+) Explicitly represents 3D shapes



3D Shape Representations: Triangle Mesh

Represent a 3D shape as a set of triangles

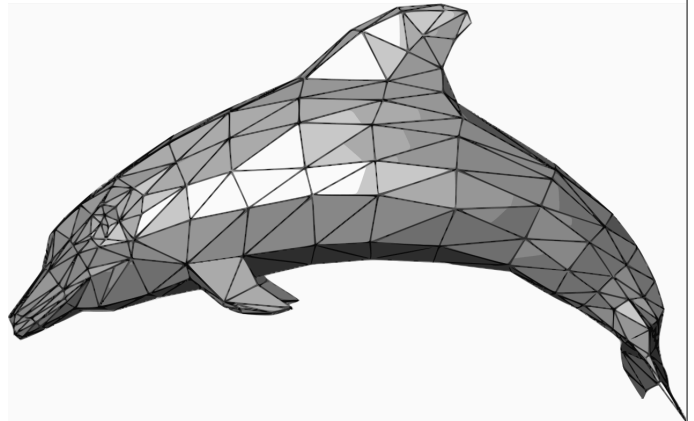
Vertices: Set of V points in 3D space

Faces: Set of triangles over the vertices

(+) Standard representation for graphics

(+) Explicitly represents 3D shapes

(+) Adaptive: Can represent flat surfaces very efficiently, can allocate more faces to areas with fine detail



Dolphin image is in the public domain

3D Shape Representations: Triangle Mesh

Represent a 3D shape as a set of triangles

Vertices: Set of V points in 3D space

Faces: Set of triangles over the vertices

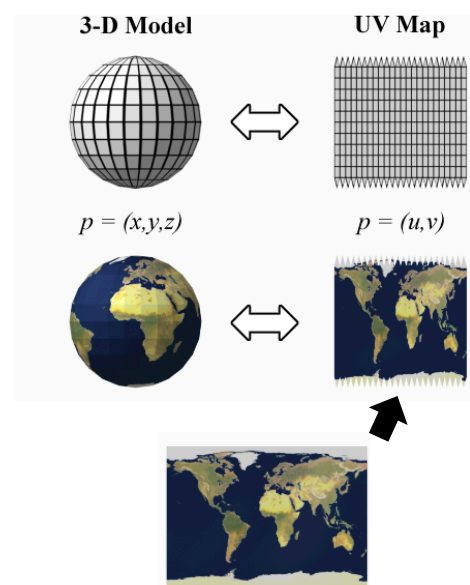
(+) Standard representation for graphics

(+) Explicitly represents 3D shapes

(+) Adaptive: Can represent flat surfaces very efficiently, can allocate more faces to areas with fine detail

(+) Can attach data on verts and interpolate over the whole surface: RGB colors, texture coordinates, normal vectors, etc.

(-) Nontrivial to process with neural nets!



UV mapping figure is licensed under CC-BY-SA 3.0. Figure slightly reorganized.

Graph-Based Representation

Mesh = Point Set + Connectivity Graph

- Non-Euclidean Domain
 - Discrete 2D Manifold
- Properties
 - Local (Vertex Neighborhoods)
no regularity
 - Global (Topology)
genus, etc
- Challenges: How to define?
 - Convolution
 - Pooling

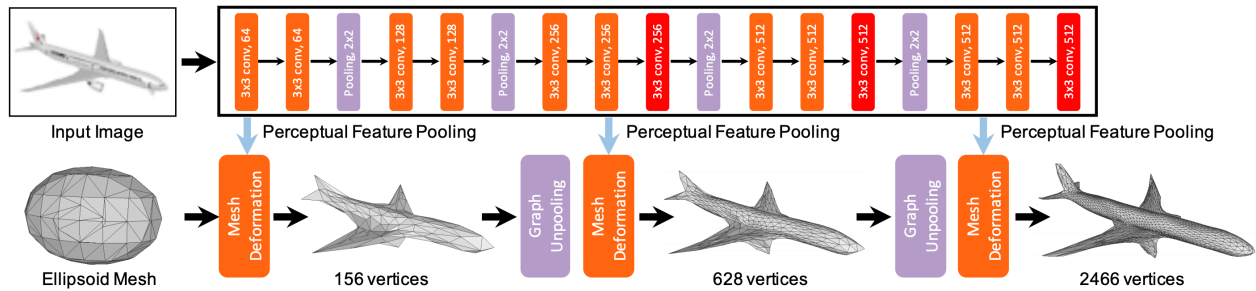


Pixel 2 Mesh

Predicting Meshes: Pixel2Mesh

Input: Single RGB
Image of an object

Output: Triangle
mesh for the object



Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

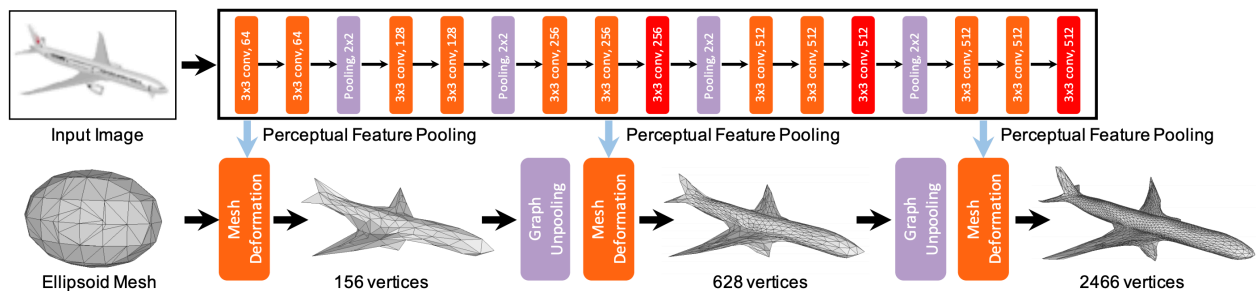
Predicting Meshes: Pixel2Mesh

Input: Single RGB
Image of an object

Key ideas:

- Iterative Refinement
- Graph Convolution
- Vertex Aligned-Features
- Chamfer Loss Function

Output: Triangle
mesh for the object



Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

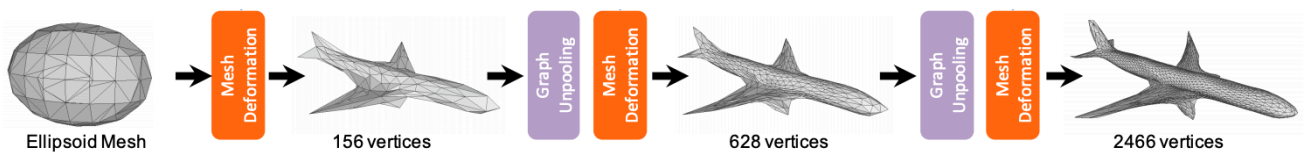
Predicting Triangle Meshes: Iterative Refinement

Idea #1: Iterative mesh refinement

Start from initial ellipsoid mesh

Network predicts offsets for each vertex

Repeat.



Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

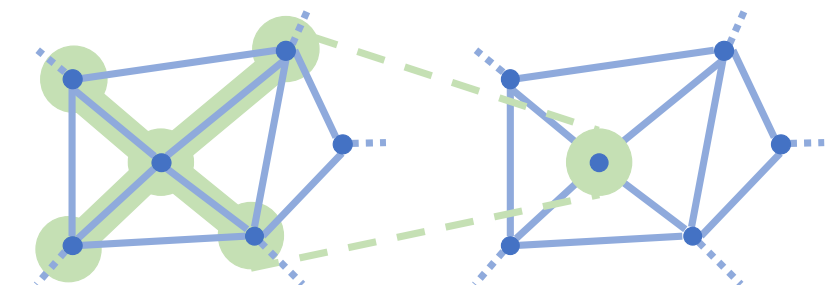
Predicting Triangle Meshes: Graph Convolution

$$f'_i = W_0 f_i + \sum_{j \in \mathcal{N}(i)} W_1 f_j$$

Vertex v_i has feature f_i

New feature f'_i for vertex v_i depends on feature of neighboring vertices $\mathcal{N}(i)$

Use same weights W_0 and W_1 to compute all outputs



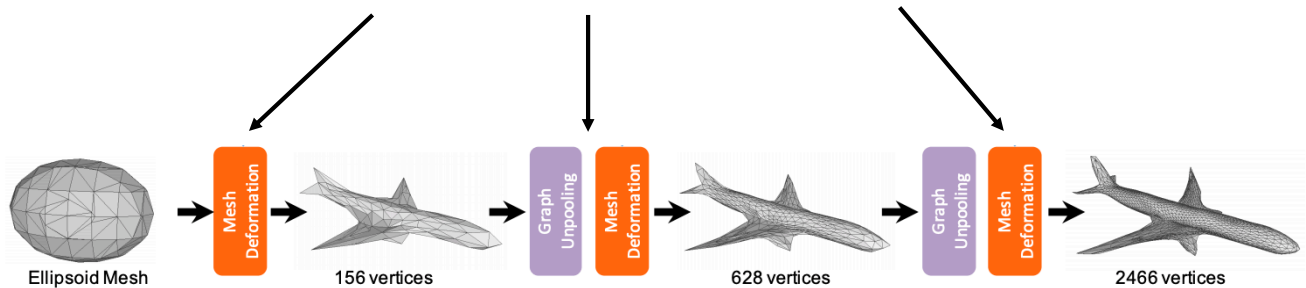
Input: Graph with a feature vector at each vertex

Output: New feature vector for each vertex

OBS: No Translation Invariance

Predicting Triangle Meshes: Graph Convolution

Each of these blocks consists of a stack of **graph convolution layers** operating on edges of the mesh

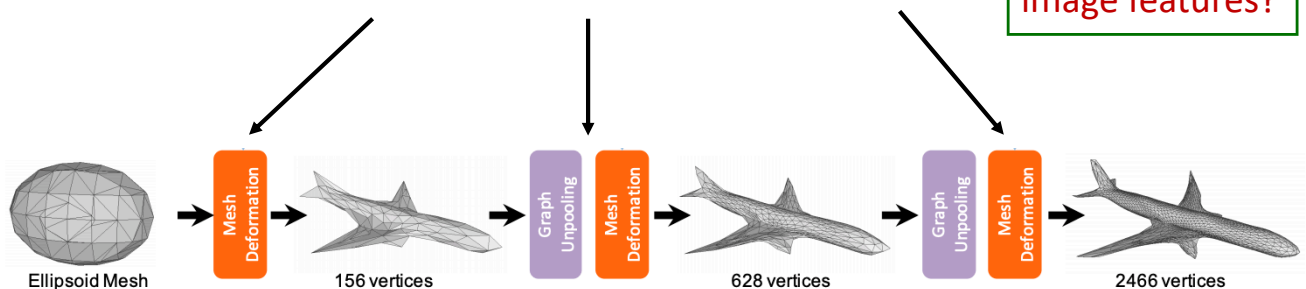


Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Triangle Meshes: Graph Convolution

Each of these blocks consists of a stack of **graph convolution layers** operating on edges of the mesh

Problem: How to incorporate image features?



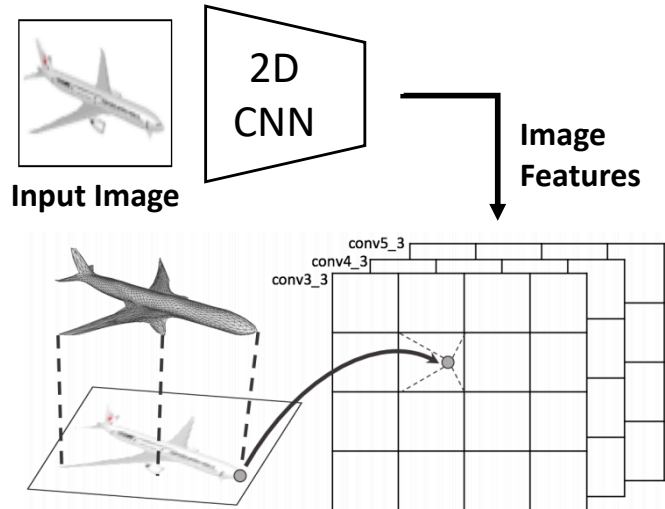
Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Triangle Meshes: Vertex-Aligned Features

Idea #2: Aligned vertex features

For each vertex of the mesh:

- Use camera information to project onto image plane
- Use bilinear interpolation to sample a CNN feature



Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

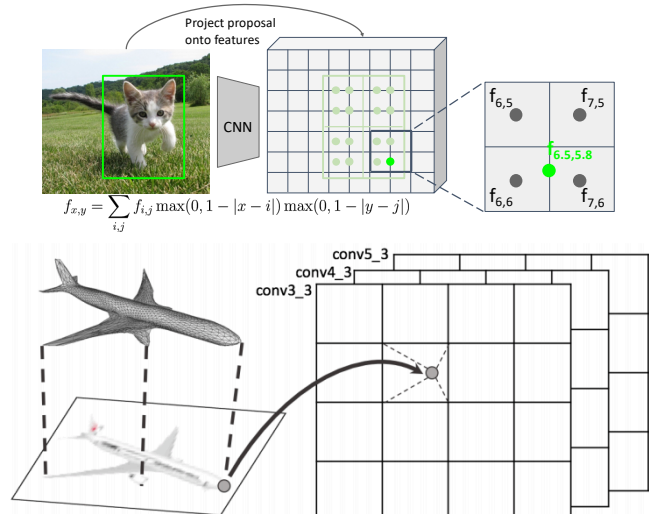
Predicting Triangle Meshes: Vertex-Aligned Features

Idea #2: Aligned vertex features

For each vertex of the mesh:

- Use camera information to project onto image plane
- Use bilinear interpolation to sample a CNN feature

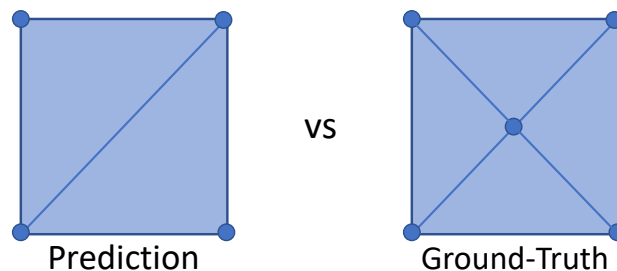
Similar to RoI-Align operation
: maintains alignment
between input image and feature
vectors



Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

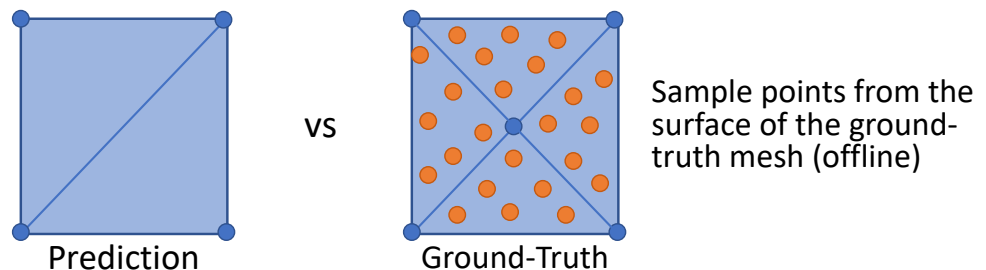


Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Idea: Convert meshes to pointclouds, then compute loss

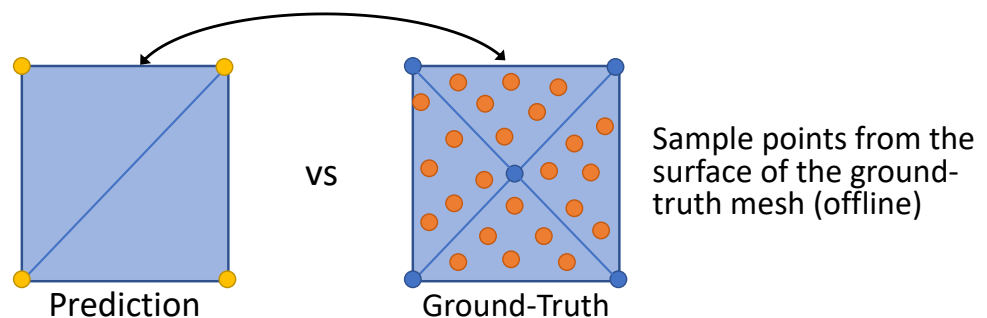


Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Loss = Chamfer distance between **predicted verts** and **ground-truth samples**



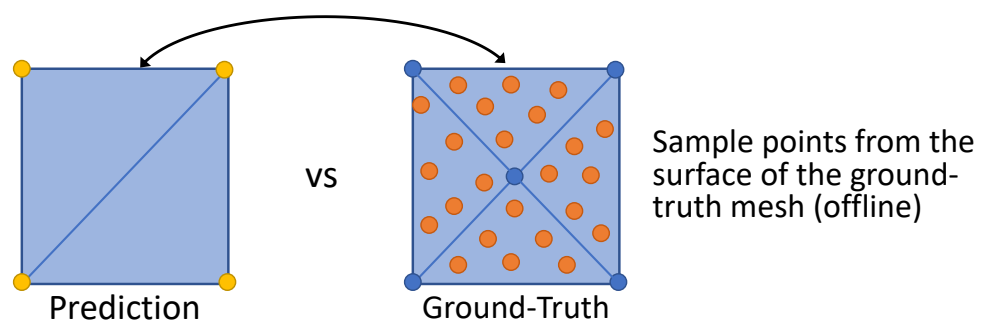
Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Loss = Chamfer distance between **predicted verts** and **ground-truth samples**

Problem: Doesn't take the interior of predicted faces into account!

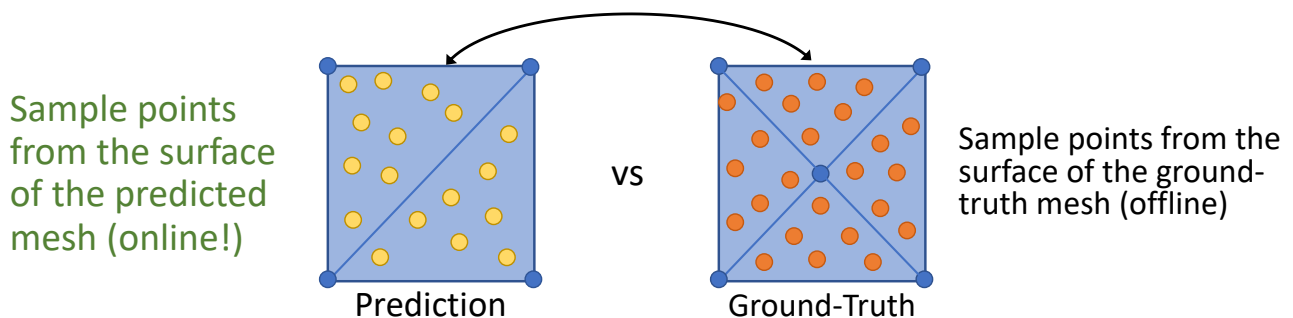


Wang et al, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images", ECCV 2018

Predicting Meshes: Loss Function

The same shape can be represented with different meshes – how can we define a loss between predicted and ground-truth mesh?

Loss = Chamfer distance between **predicted samples** and **ground-truth samples**



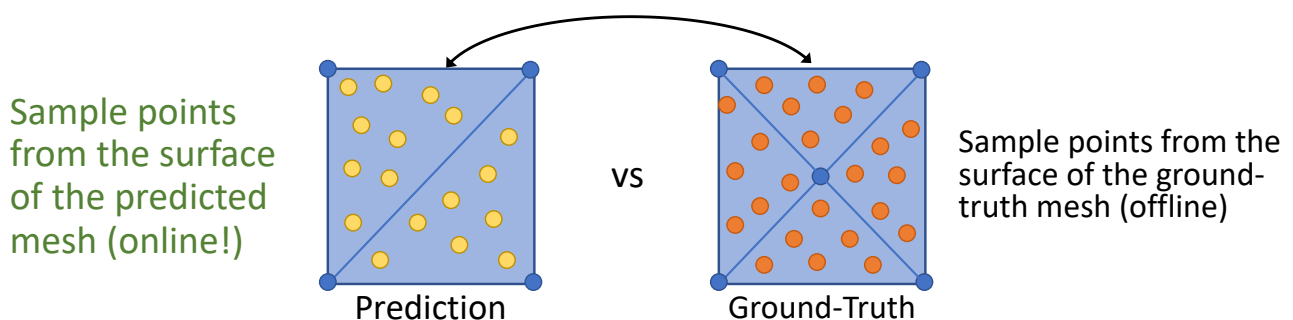
Smith et al, "GEOMETRICS: Exploiting Geometric Structure for Graph-Encoded Objects", ICML 2019

Predicting Meshes: Loss Function

Problem: Need to sample online! Must be efficient!

Problem: Need to backprop through sampling!

Loss = Chamfer distance between **predicted samples** and **ground-truth samples**



Smith et al, "GEOMETRICS: Exploiting Geometric Structure for Graph-Encoded Objects", ICML 2019

Loss & Regularization

- Losses

- Chamfer Loss: $l_c = \sum_p \min_q \|p - q\|_2^2 + \sum_q \min_p \|p - q\|_2^2$.

- Normal Loss: $l_n = \sum_p \sum_{q=\arg \min_q (\|p-q\|_2^2)} \|\langle p - k, \mathbf{n}_q \rangle\|_2^2$, s.t. $k \in \mathcal{N}(p)$

- Regularization

- Laplacian: $\delta_p = p - \sum_{k \in \mathcal{N}(p)} \frac{1}{\|\mathcal{N}(p)\|} k$,

$$l_{lap} = \sum_p \|\delta'_p - \delta_p\|_2^2,$$

- Edge Length: $l_{loc} = \sum_p \sum_{k \in \mathcal{N}(p)} \|p - k\|_2^2$.

$l_{all} = l_c + \lambda_1 l_n + \lambda_2 l_{lap} + \lambda_3 l_{loc}, \quad \lambda_1 = 1.6e - 4, \lambda_2 = 0.3 \text{ and } \lambda_3 = 0.1$

Mesh R-CNN

3D Shape Prediction: Mesh R-CNN

Mask R-CNN:

2D Image -> 2D shapes



He, Gkioxari, Dollár, and Girshick, "Mask R-CNN", ICCV 2017

Mesh R-CNN:

2D Image -> **Triangle Meshes**



Gkioxari, Malik, and Johnson, "Mesh R-CNN", ICCV 2019

Mesh R-CNN: Task

Input: Single RGB image

Output:

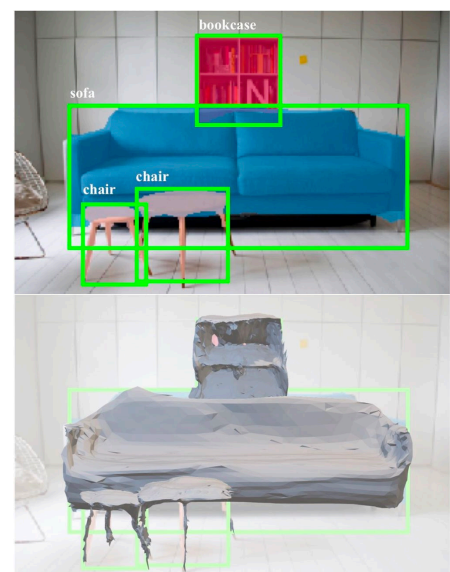
A set of detected objects
For each object:

- Bounding box
- Category label
- Instance segmentation
- 3D triangle mesh

Mask R-CNN

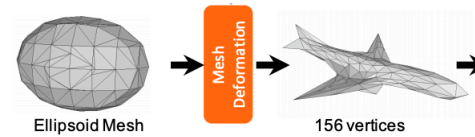
Mesh head

OBS: Arbitrary Topology

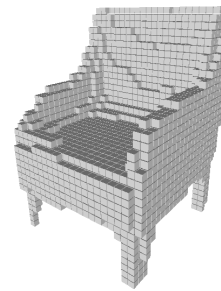


Mesh R-CNN: Hybrid 3D shape representation

Mesh deformation gives good results, but the topology (verts, faces, genus, connected components) fixed by the initial mesh



Our approach: Use voxel predictions to create initial mesh prediction!

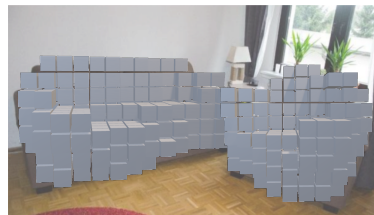


Mesh R-CNN Pipeline

Input image



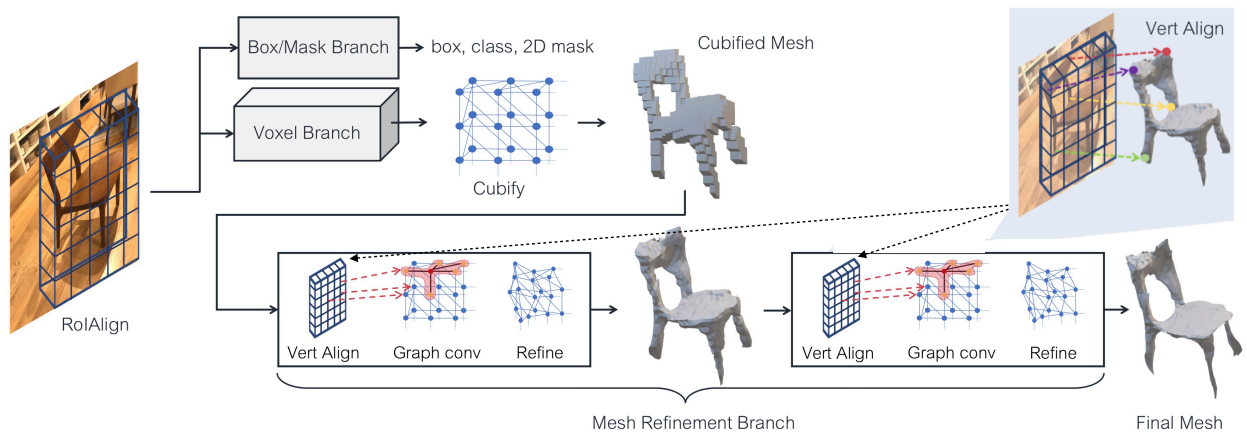
2D object recognition



3D object meshes

3D object voxels

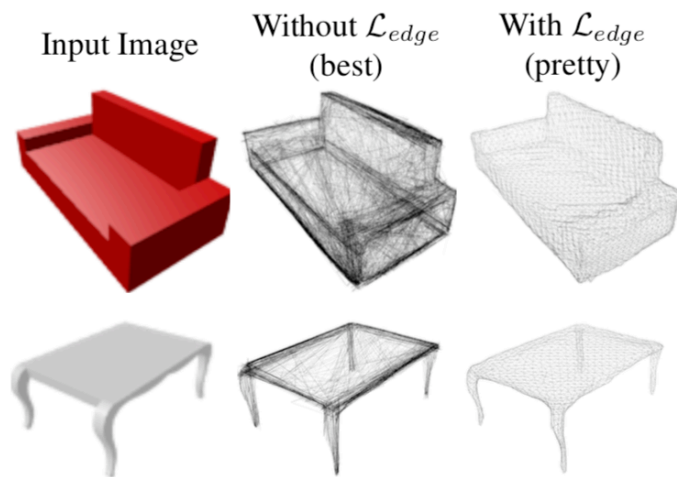
Mesh R-CNN Architecture



Mesh R-CNN: ShapeNet Results

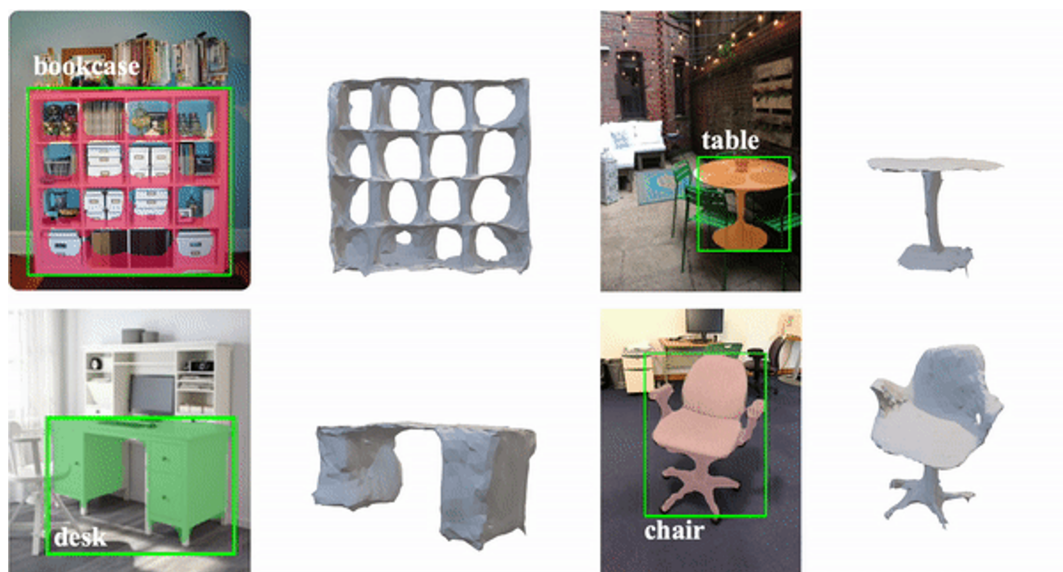


Mesh R-CNN: Shape Regularizers



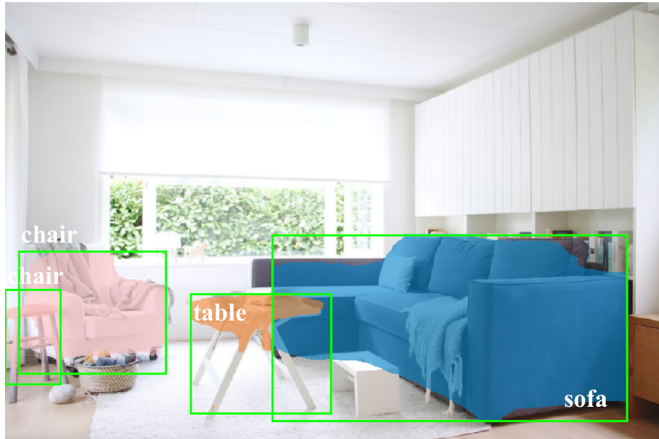
Using Chamfer as only mesh loss gives degenerate meshes. Also need "mesh regularizer" to encourage nice predictions: $\mathcal{L}_{edge}$ = minimize L2 norm of edges in the predicted mesh

Mesh R-CNN: Pix3D Results

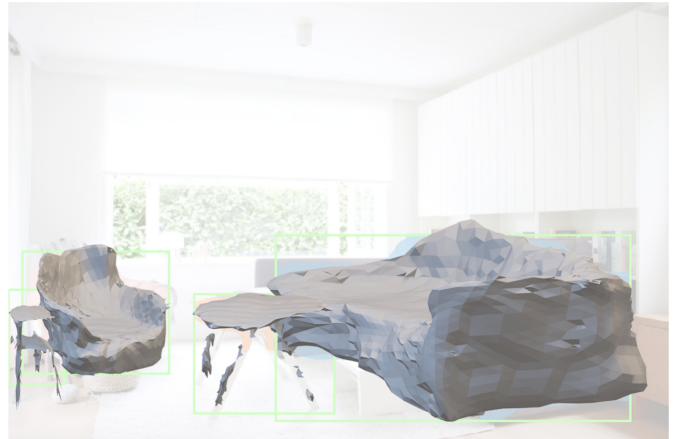


Mesh R-CNN: Pix3D Results

Predicting many objects per scene



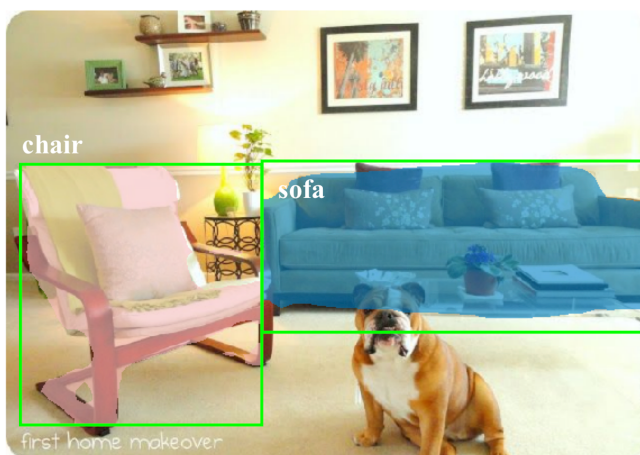
Box & Mask Predictions



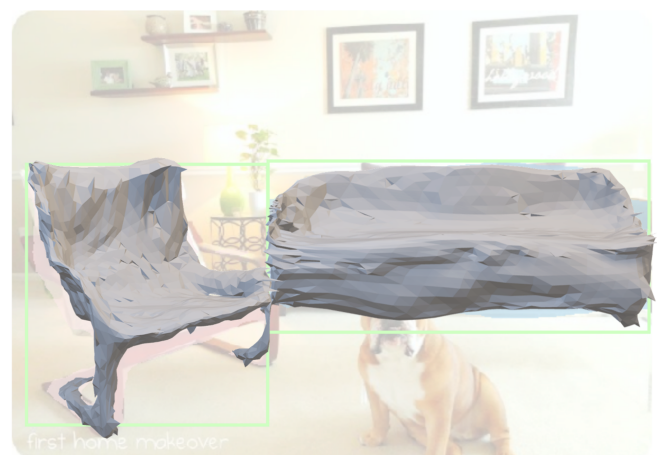
Mesh Predictions

Mesh R-CNN: Pix3D Results

Amodal completion: predict occluded parts of objects



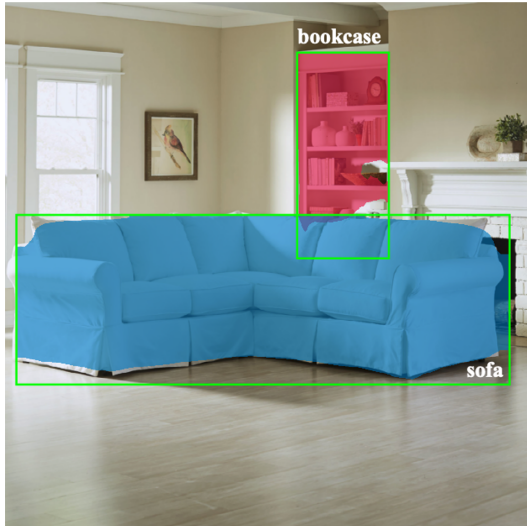
Box & Mask Predictions



Mesh Predictions

Mesh R-CNN: Pix3D Results

Segmentation failures
propagate to meshes



Box & Mask Predictions



Mesh Predictions