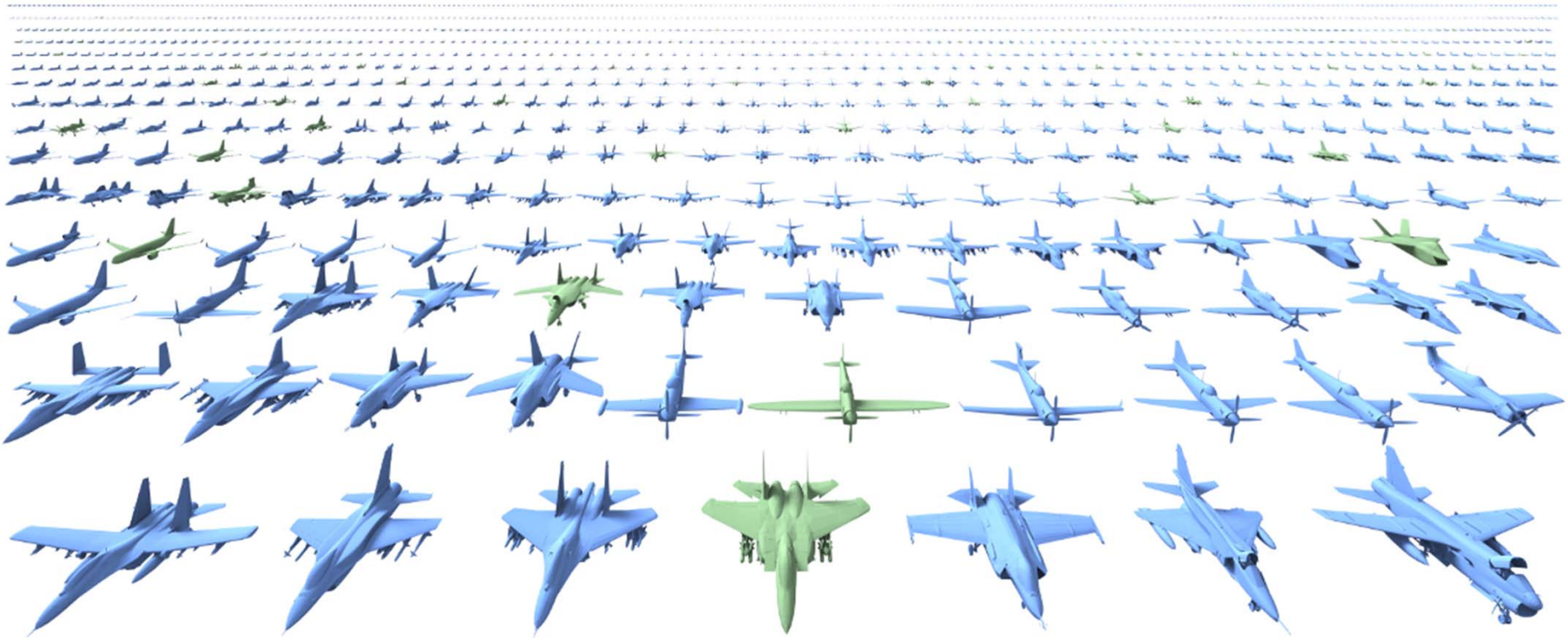
Generative models of 3D Shapes and Scenes



Evangelos Kalogerakis – 574/674

What representation should we generate?

- **Multiple views**


3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks, Lun et al, 2017

- **Point cloud**

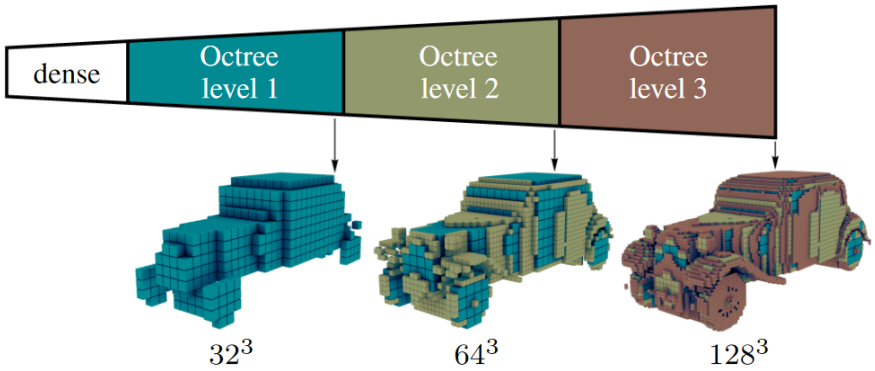

A Point Set Generation Network for 3D Object Reconstruction from a Single Image, Fan et al 2017

- **Voxels**

Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, Wu et al 2017

What representation should we generate?

- **Octree cells**



Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs, Tatarchenko et al. 2017

- **Parametric Patches**

AtlasNet, Groueix et al. 2018

- **Volumetric Primitives**

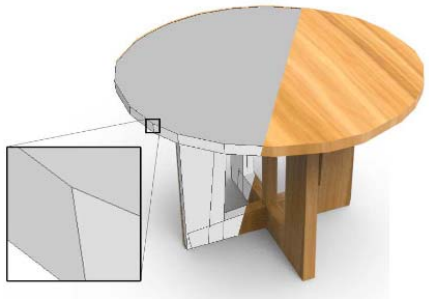
Learning Shape Abstractions by Assembling Volumetric Primitives Tulsiani et al. 2017

What representation should we generate?

- **Implicits**

Convolutional Occupancy Networks
Peng al 2020

- **Meshes**



BSP-Net, Chen et al. 2020

- **Structure/Program**

ShapeAssembly, Jones et al 2020

Neural Program Generator

Neural Program Executor

```
Draw("Top", "Circle", position, geometry)

for(i < 2, "translation", a)
  for(j < 2, "translation", b)
    Draw("Leg", "Cub", position + i*a + j*b, geometry)

for(i < 2, "translation", c)
  Draw("Layer", "Rec", position + i*c, geometry)
```

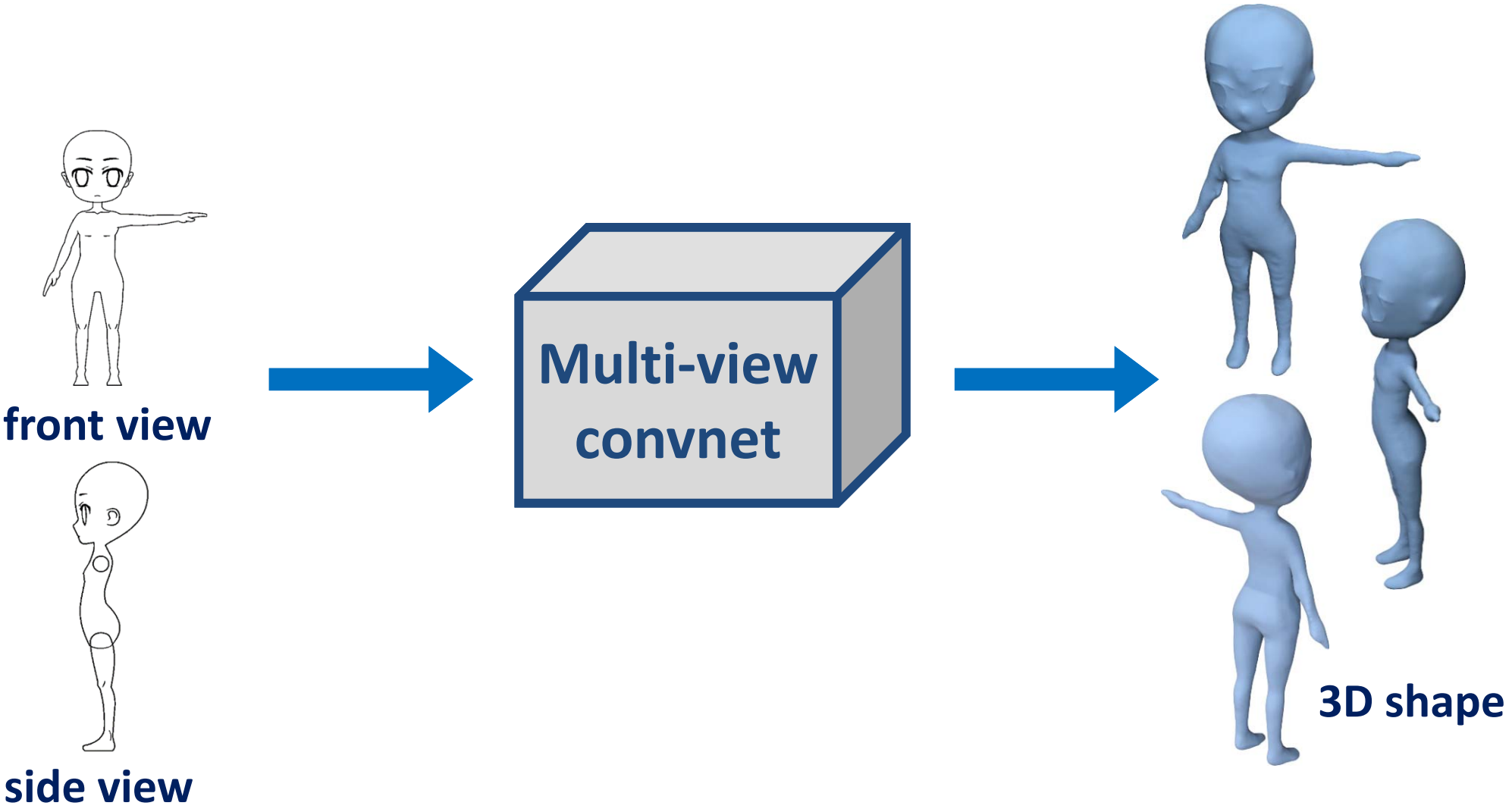
How to generate shapes/scenes?

- Encoder-Decoders
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

How to generate shapes/scenes?

- **Encoder-Decoders**
  - **Case Study: Multi-view decoder**
  - Case Study: Implicit decoder
  - Case Study: Patch decoder
  - Case Study: Mesh Decoder
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

Sketch-to-shape

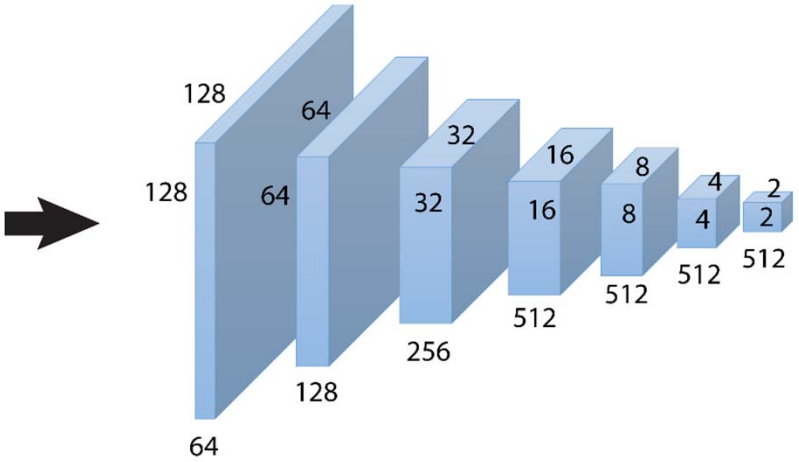


3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks, Lun et al. 2017

Encoder

front view

side view



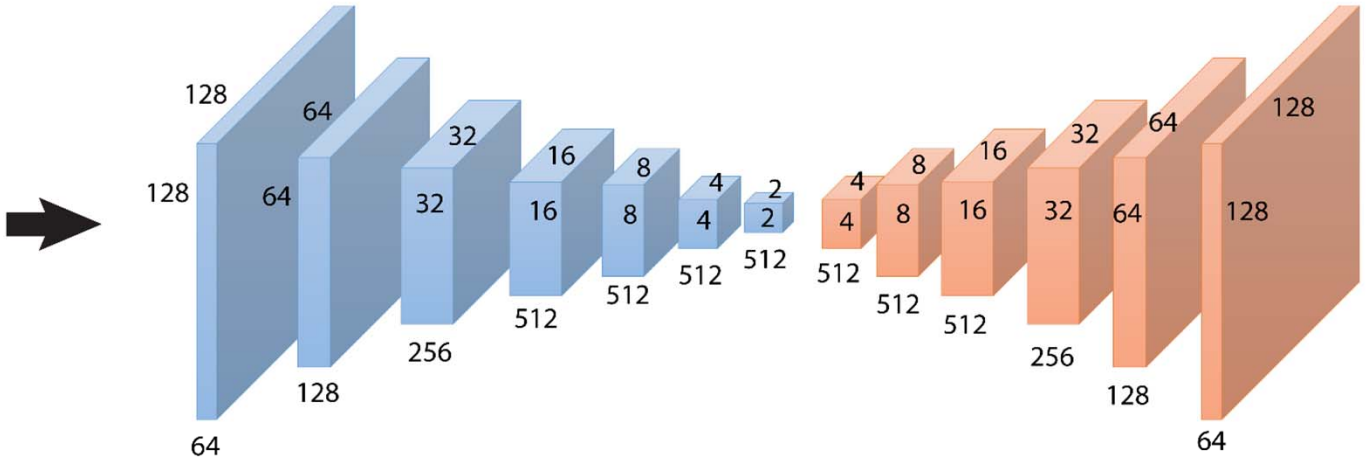
Feature representations capturing **increasingly larger context** in the sketches

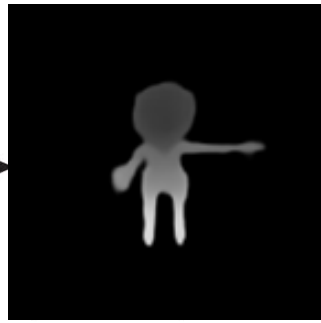
Decoder

Infer **depth and normal maps**

front view

side view



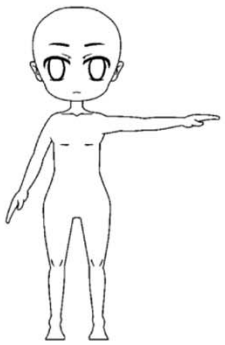


depth map

Feature representations generating shape information **at increasingly finer scales**

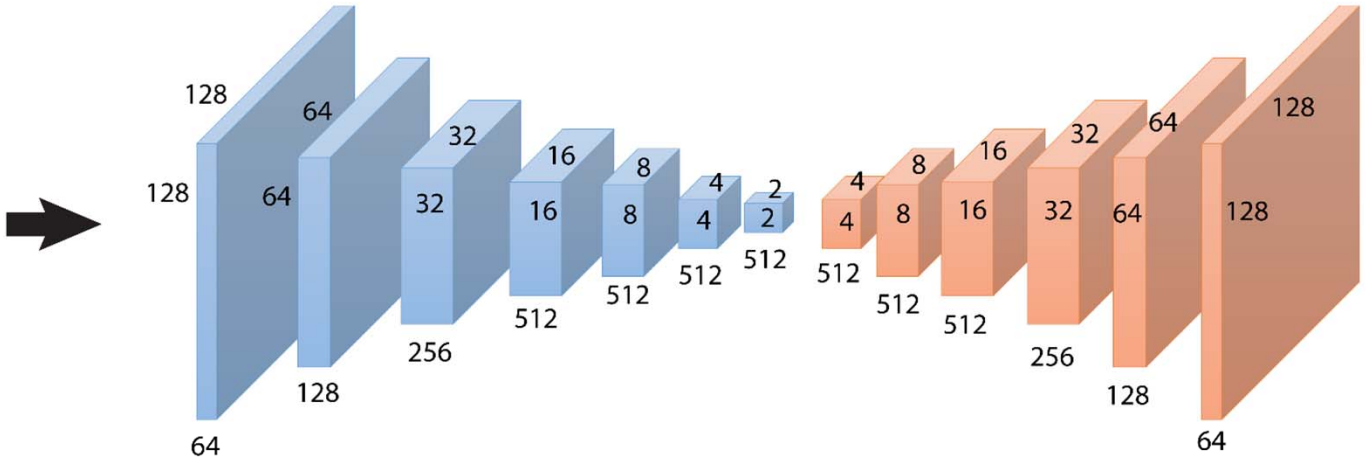
Decoder

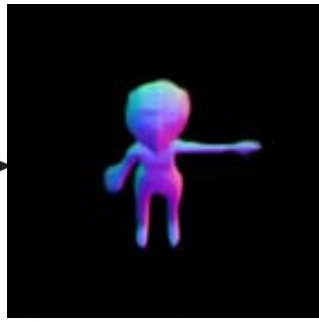
Infer **depth and normal maps**



front view

side view



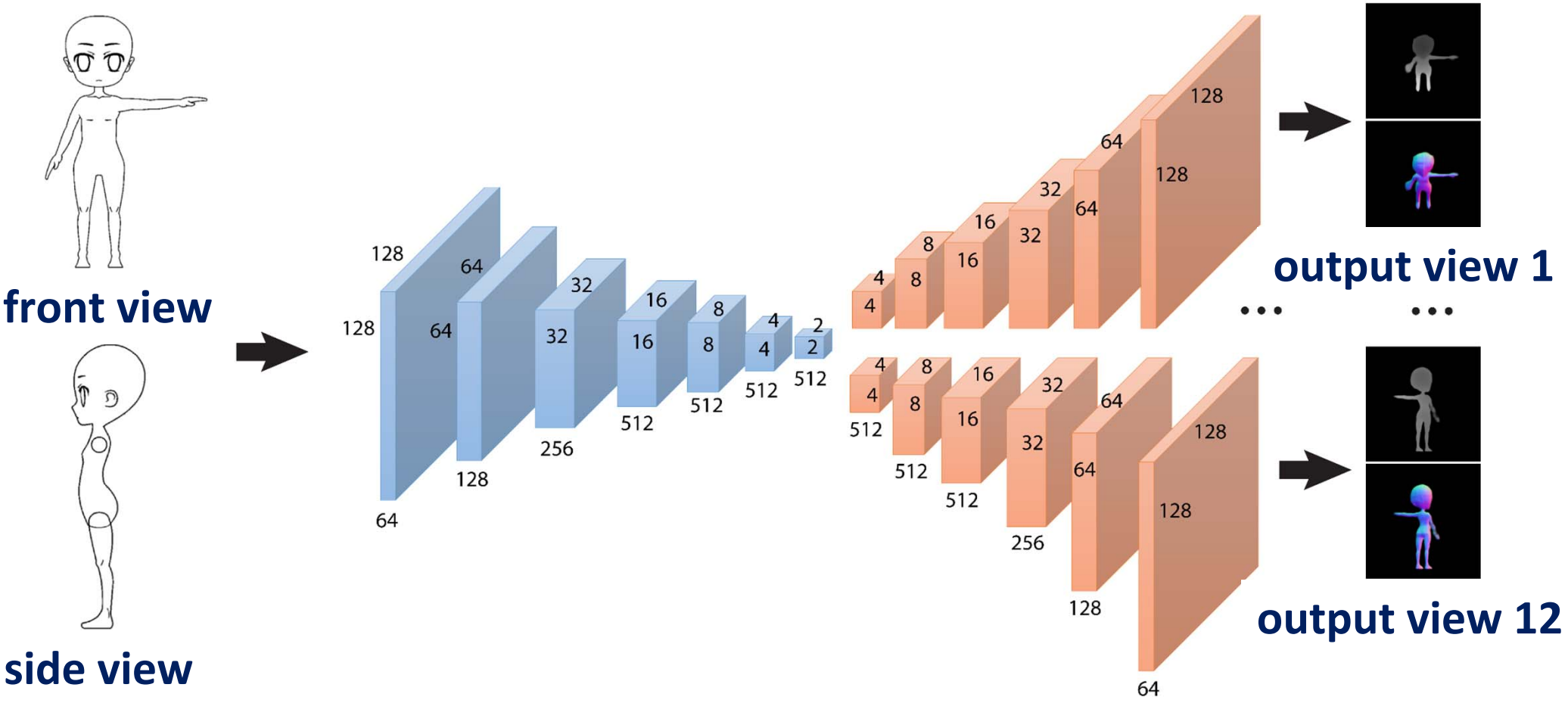


+normal map

Feature representations generating shape information **at increasingly finer scales**

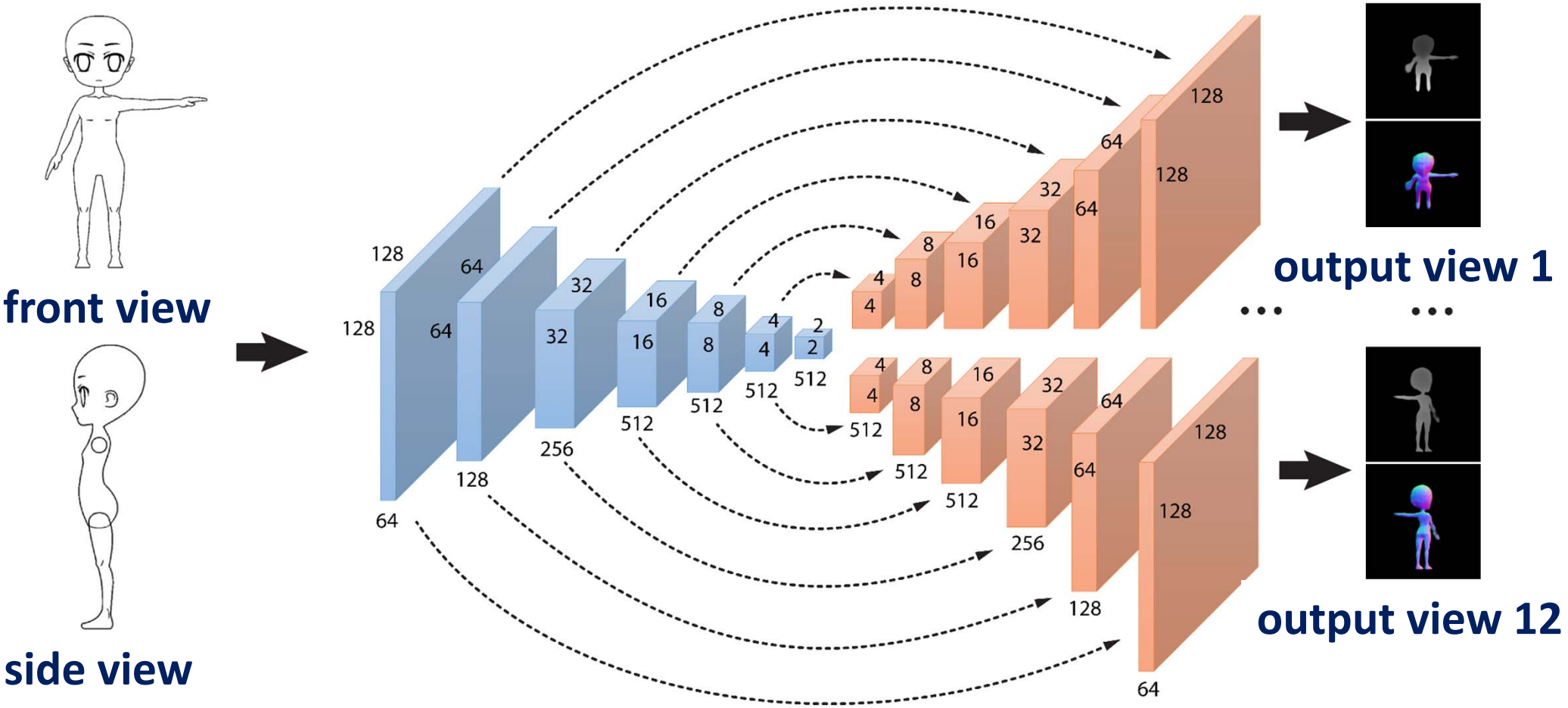
Multi-view decoder

Infer depth and normal maps **for several views**



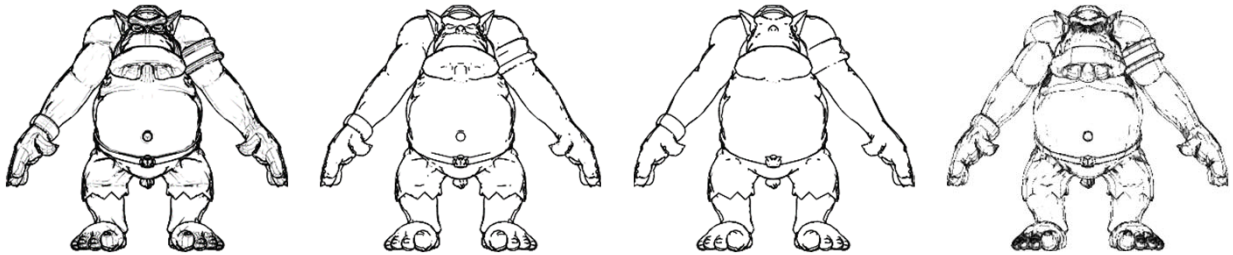
“U-Net”

Feature representations in the decoder depend on **previous layer & encoder’s corresponding layer**



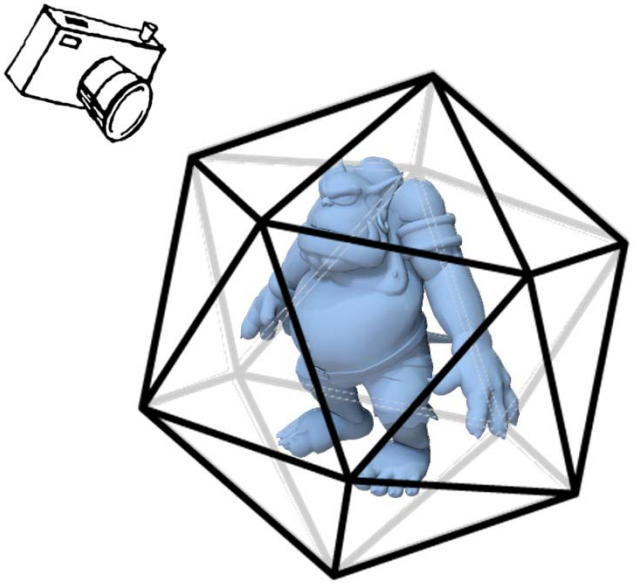
U-net: Ronneberger et al. 2015,
Isola et al. 2016

Training data

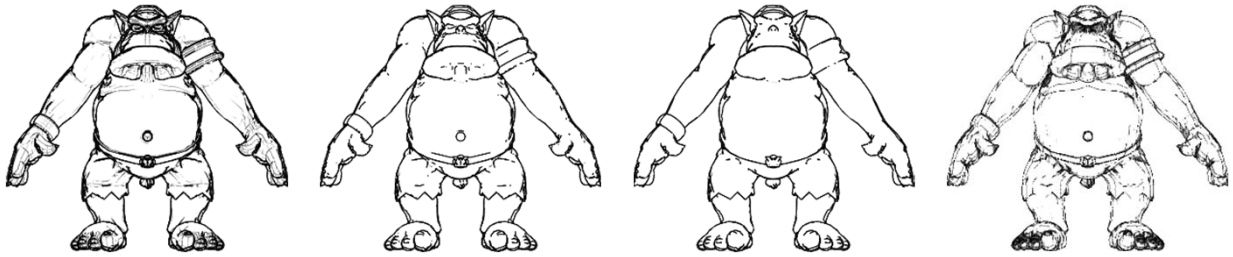


Synthetic line drawings

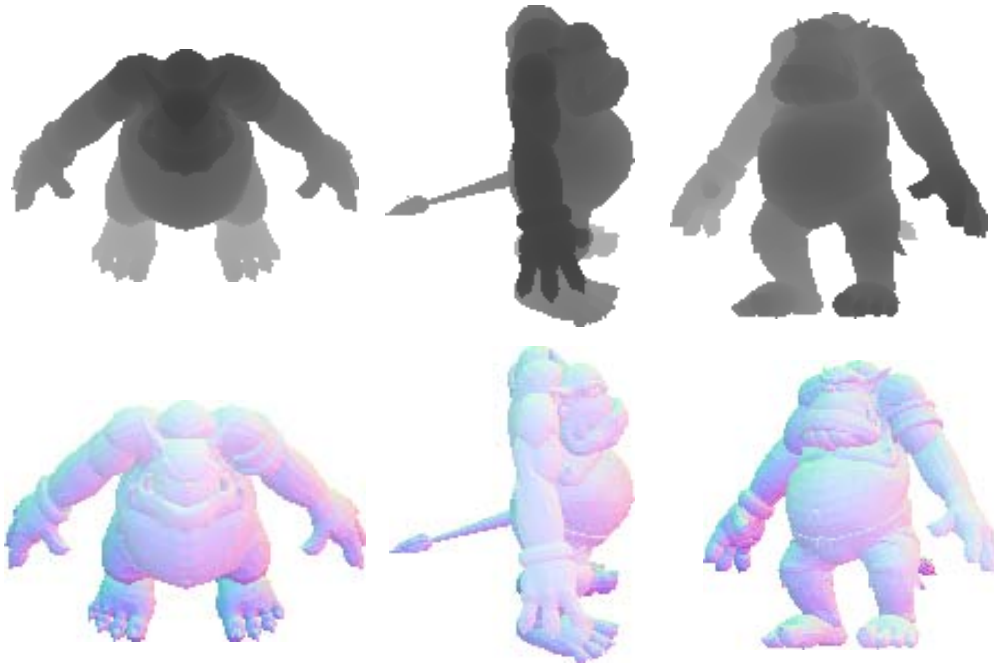
Training data



12 views



Synthetic line drawings

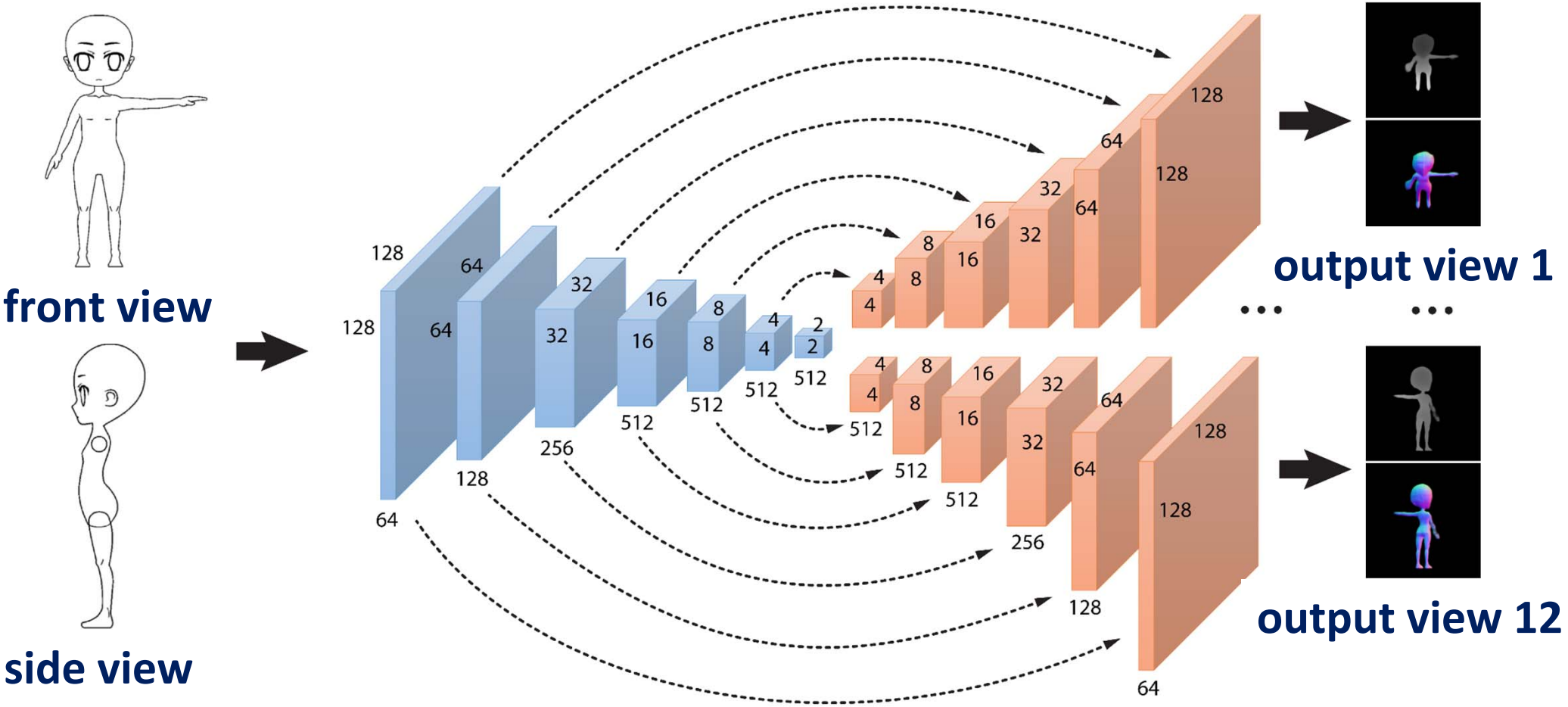


Training depth and normal maps

Initial training loss

Penalize per-pixel depth reconstruction error: $\sum_{pixels} | d_{pred} - d_{gt} |$

& per-pixel normal reconstruction error: $\sum_{pixels} (1 - \mathbf{n}_{pred} \cdot \mathbf{n}_{gt})$



U-net: Ronneberger et al. 2015,
Isola et al. 2016

Training data

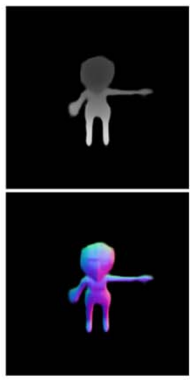
Character
10K models

Chair
10K models

Airplane
3K models

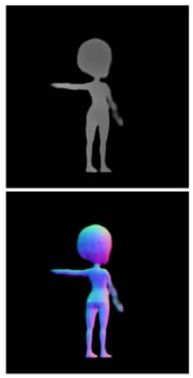
Models from “The Models Resource” &
3D Warehouse

Multi-view depth & normal map fusion



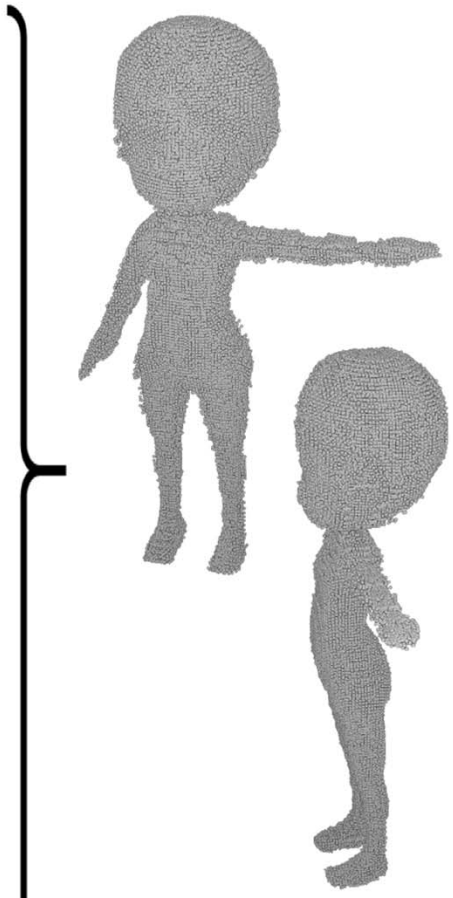
output view 1

...



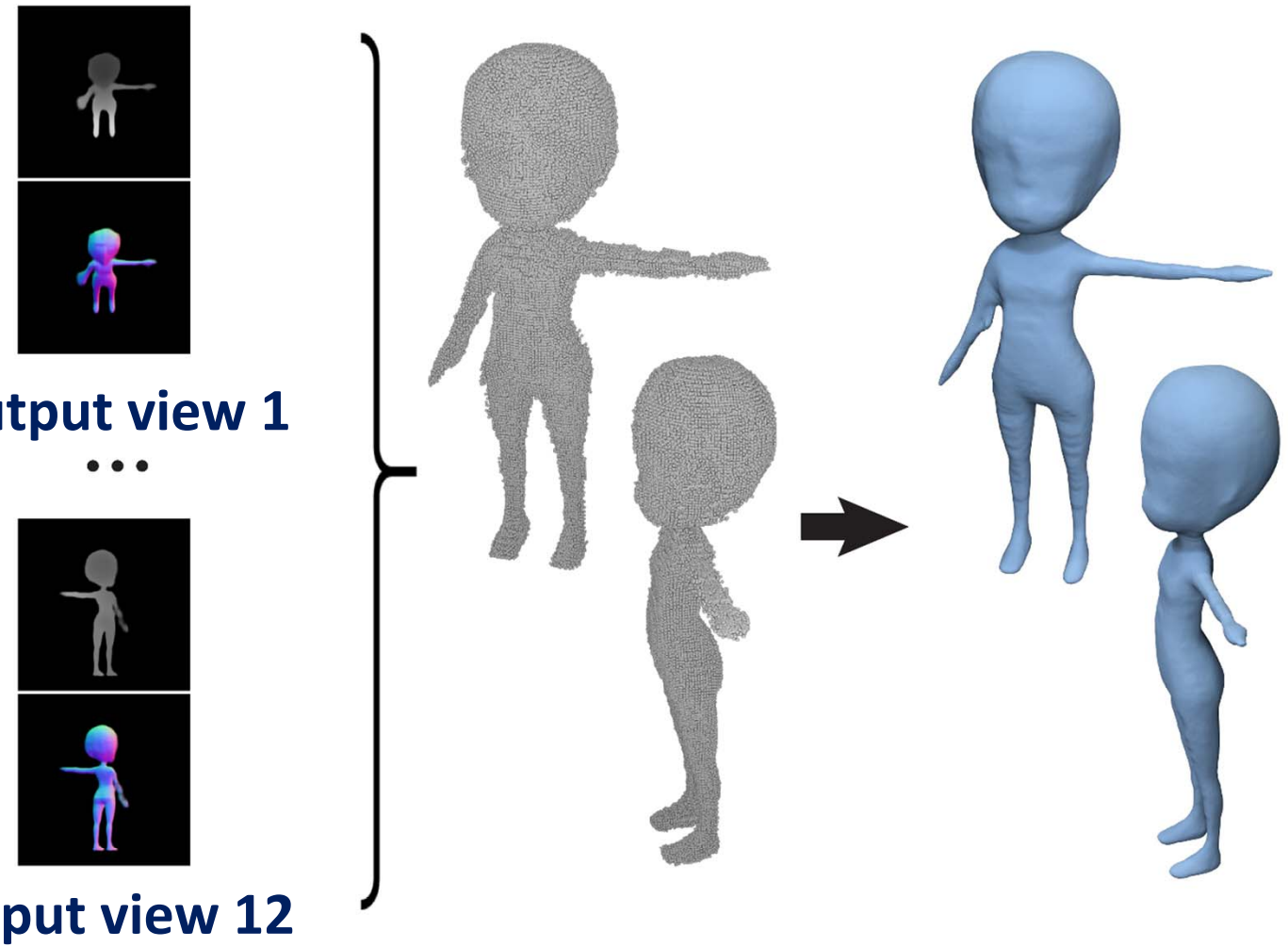
output view 12

Multi-view depth & normal maps

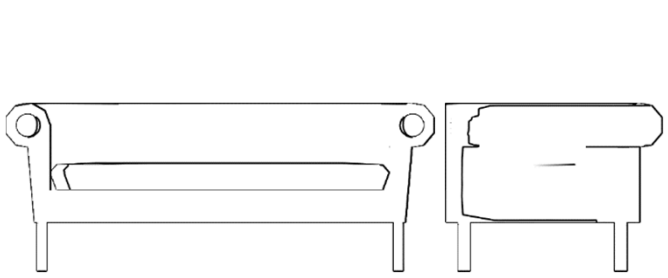


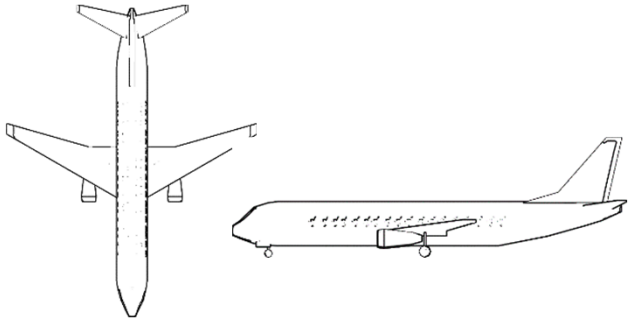
Consolidated point cloud

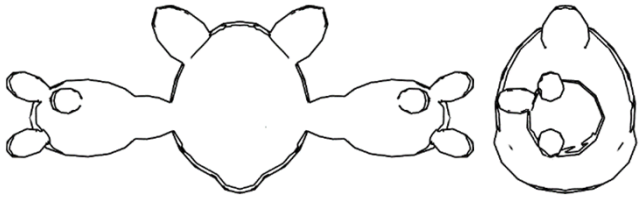
Surface reconstruction

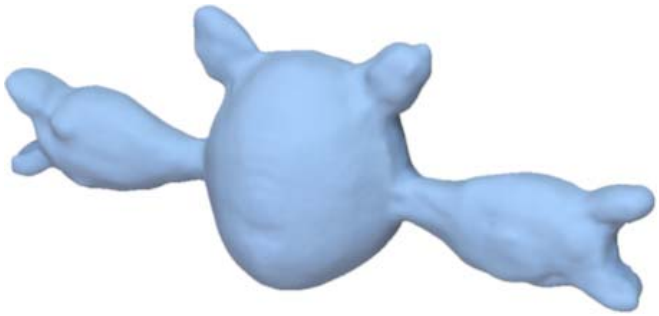


Qualitative results







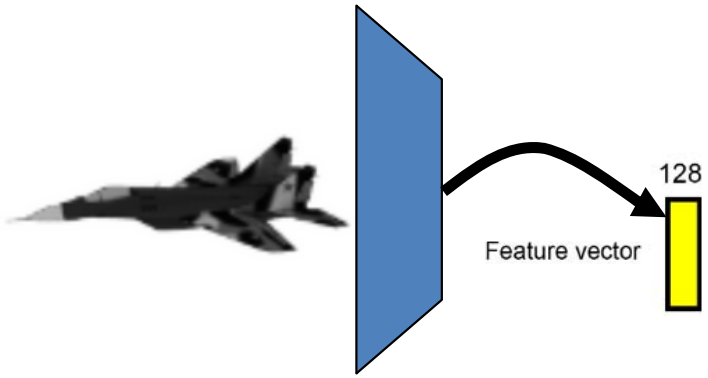


How to generate shapes/scenes?

- **Encoder-Decoders**
  - Case Study: Multi-view decoder
  - **Case Study: Implicit decoder**
  - Case Study: Patch decoder
  - Case Study: Mesh Decoder
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

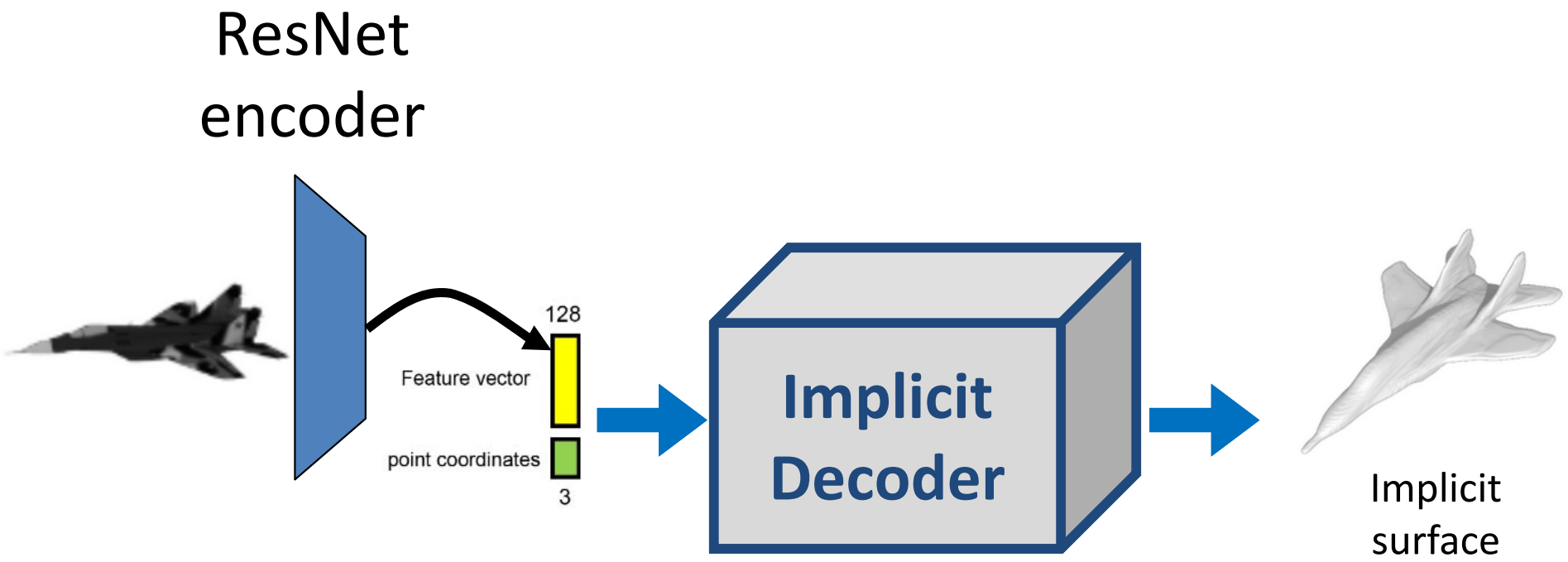
Image-to-shape

ResNet
encoder



Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Image-to-shape



Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Implicit surfaces

Surface is defined by **implicit** function f(x,y,z) = 0. In other words, implicit surfaces are isosurfaces through some scalar field in 3D.

3-D example

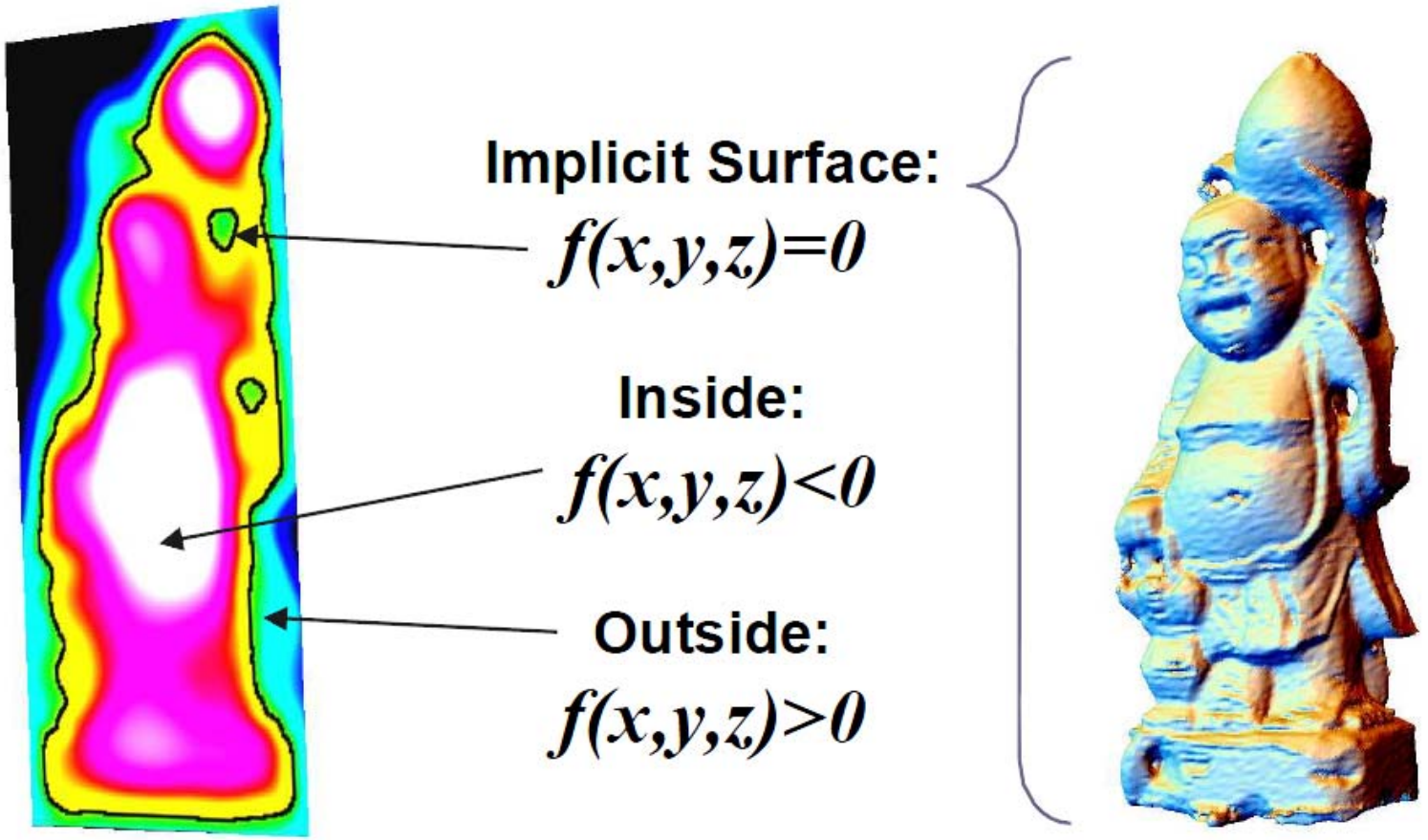
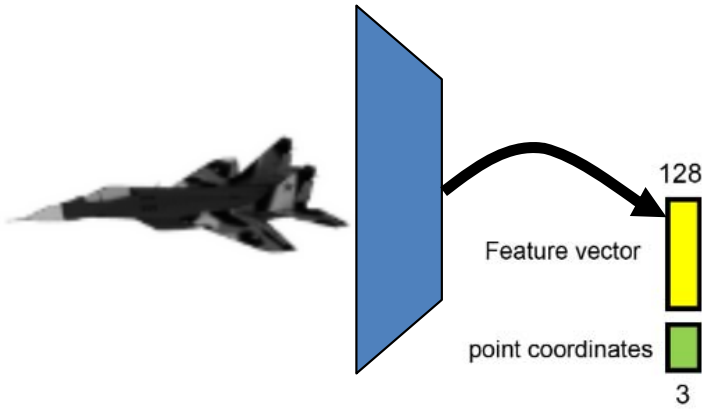
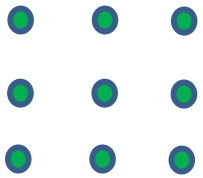


Image-to-shape

ResNet encoder



At test time, pass the coordinates of each point in a R^3 grid

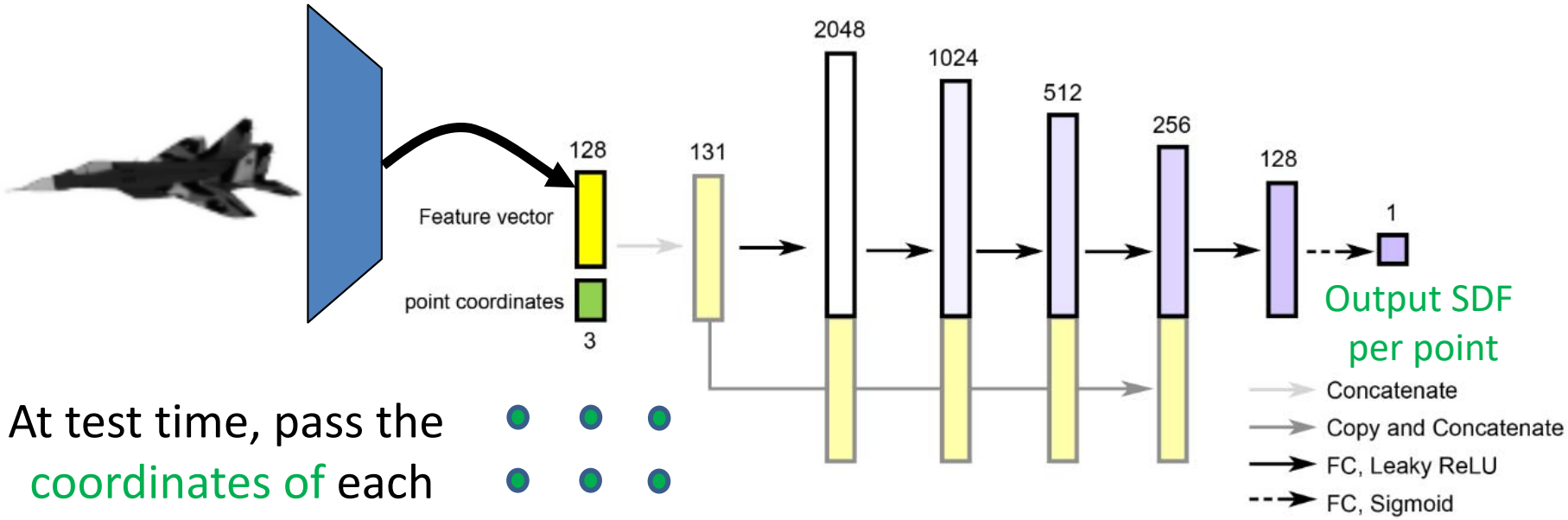


Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Image-to-shape

ResNet encoder

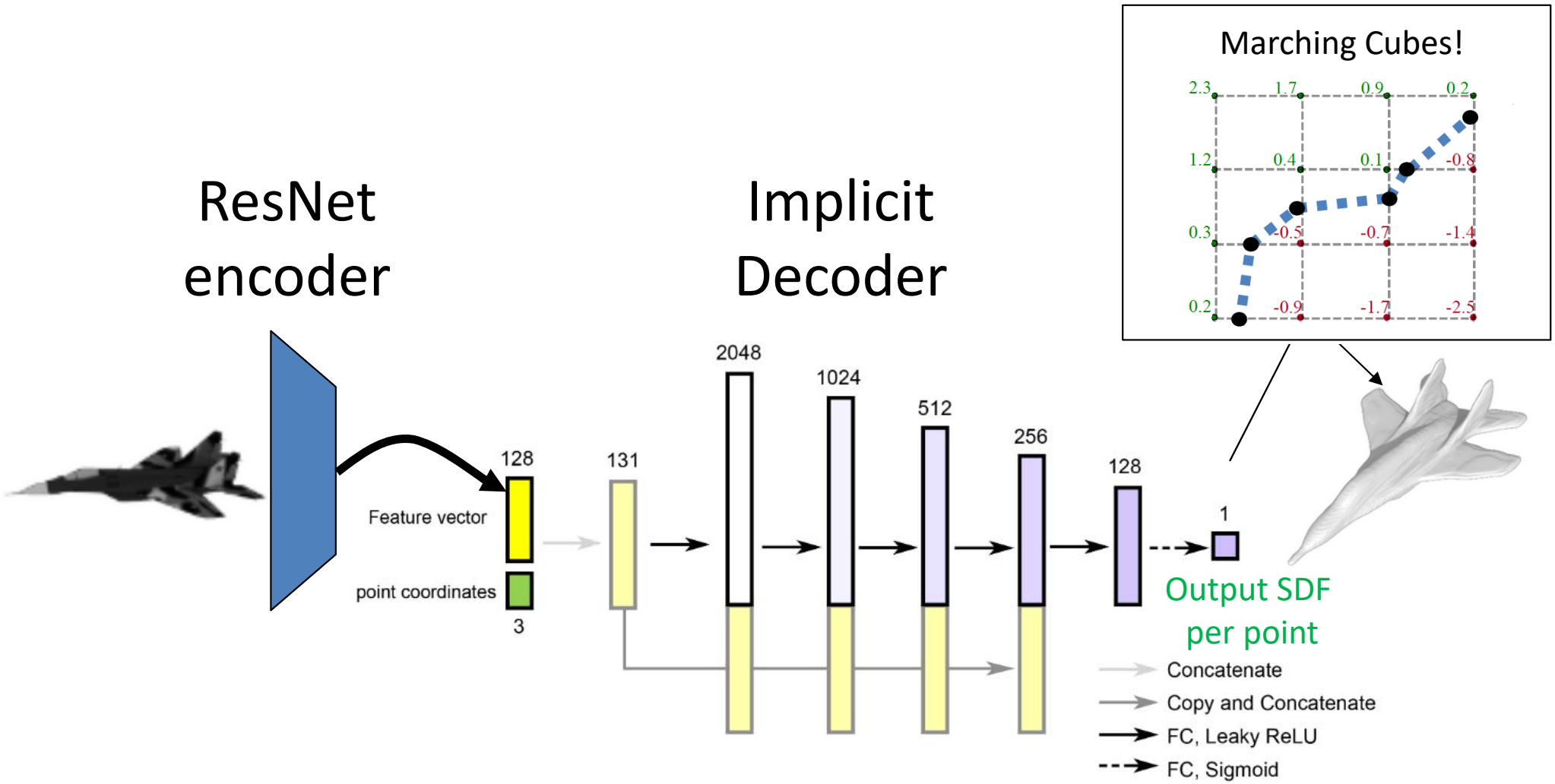
Implicit Decoder



At test time, pass the coordinates of each point in a R^3 grid

Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Image-to-shape



Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Training

Given a sample point $\boldsymbol{p_i}$' , the network output $f(\boldsymbol{p_i}')$, and its ground-truth SDF value s_i:

$$L(\ f(\mathbf{p_i}'),\ s_i\ ) = \left|\ f(\mathbf{p_i}') - \ s_i\ \right|^2$$

One option is to use L2 loss (summed over sample points)

Training

Given a sample point $\boldsymbol{p_i}$', the network output $f(\boldsymbol{p_i}')$, and its ground-truth SDF value s_i:

$$L(\ f(\mathbf{p_i}'),\ s_i\ ) = \left|\ f(\mathbf{p_i}') - \ s_i\ \right|^2$$

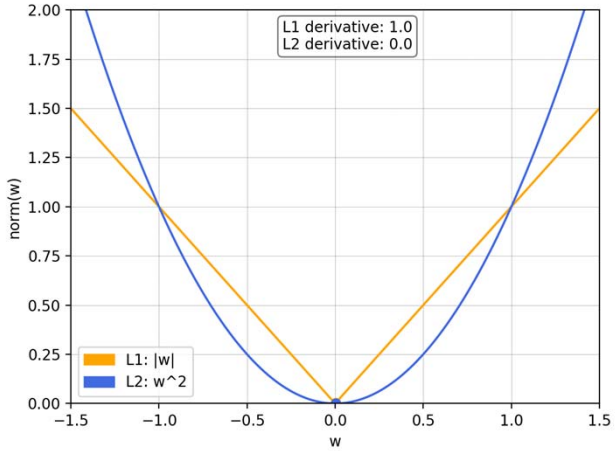
One option is to use L2 loss (summed over sample points)

To create sample points, take each surface point and perturb it along its normal (e.g., use Gaussian for perturbing)

$$\mathbf{p_i}' = \mathbf{p_i} + \varepsilon \cdot \mathbf{n}_i$$

$$(\text{where } \mathbf{p_i} \text{ is a surface point, and } \mathbf{n}_i \text{ is its normal})$$

Training



Another option is to use L1 loss:

$$L(\ f(\mathbf{p_i}'),\ s_i\ ) = \left|\ f(\mathbf{p_i}') - \ s_i\ \right|$$

Better results are achieved with the use of **clamped L1 loss** (focus on predicting correct SDFs near the surface)

$$L(\ f(\mathbf{p_i}'),\ s\ ) = \left|\ clamp(\ f(\mathbf{p_i}'),\ \sigma\ ) - \ \ clamp(\ s_i,\ \sigma\ )\right|$$

$$where\ \ clamp(\ t,\ \delta\ ) = \min(\ \delta,\ \max(-\delta, t)\ )$$

Results

Input image

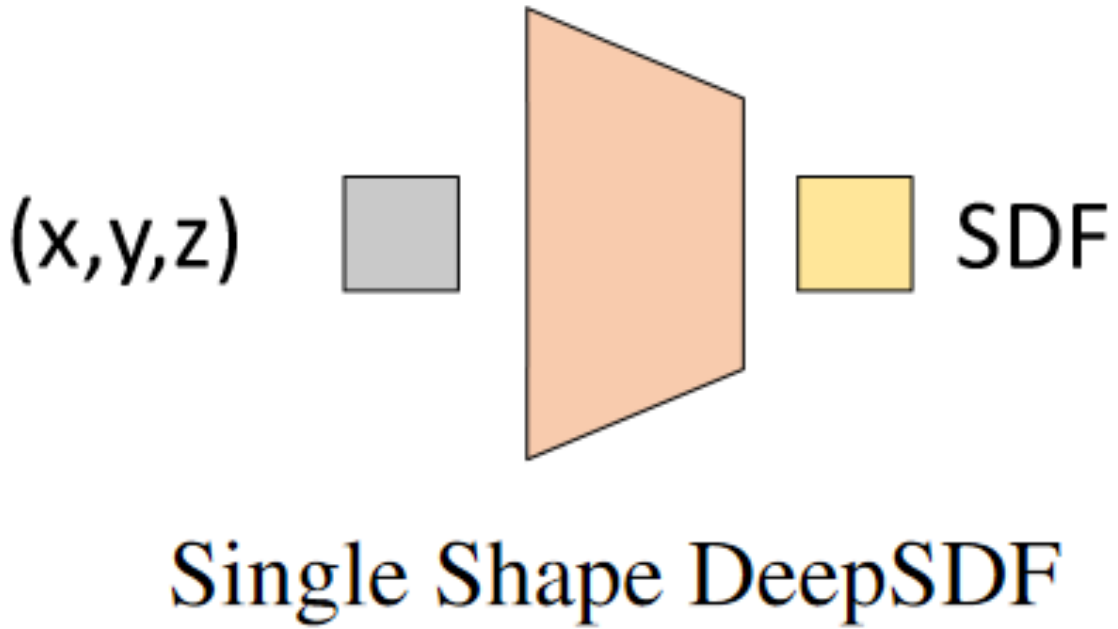
IM-SVR
(implicit decoder)

Ground truth

Learning Implicit Fields for Generative Shape Modeling, Chen al. 2019

Assignment 4

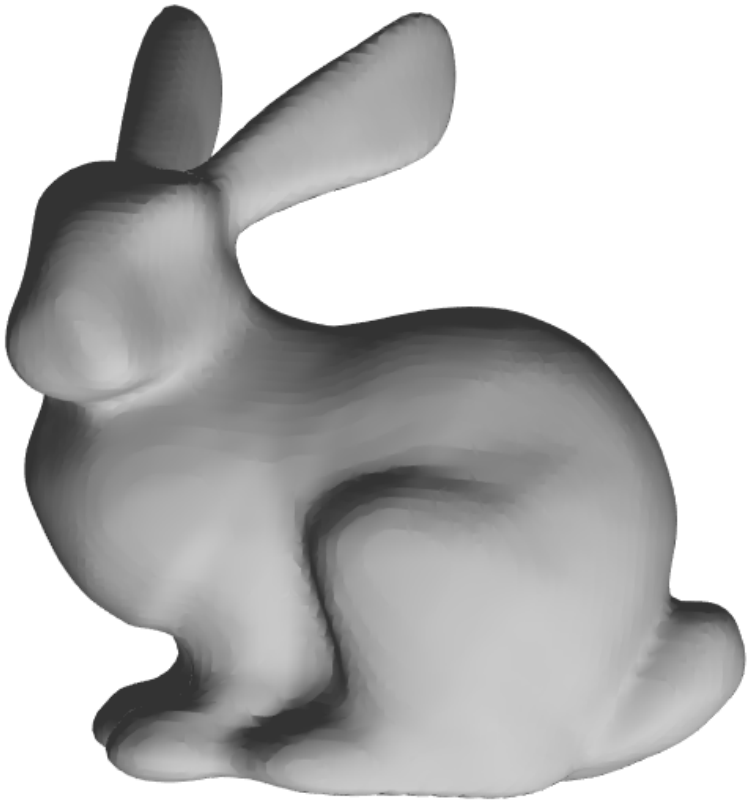
A simpler version where you train a SDF neural network on a single point cloud (like in assignment 1)



Single Shape DeepSDF

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

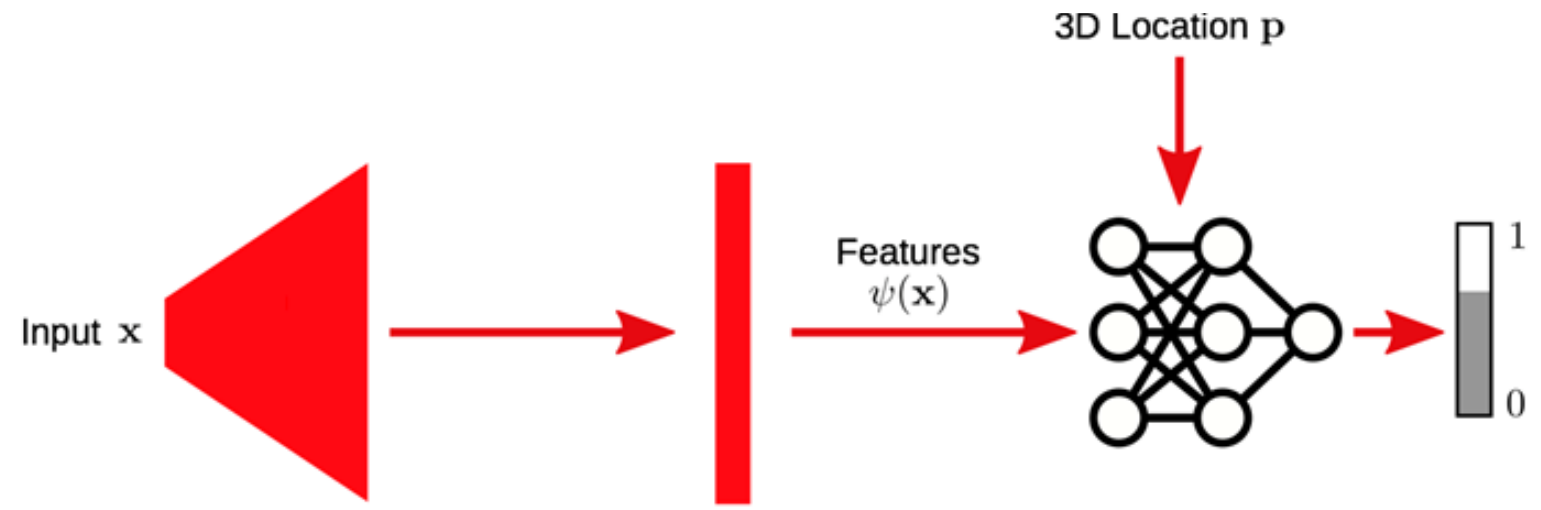
DeepSDF vs MLS



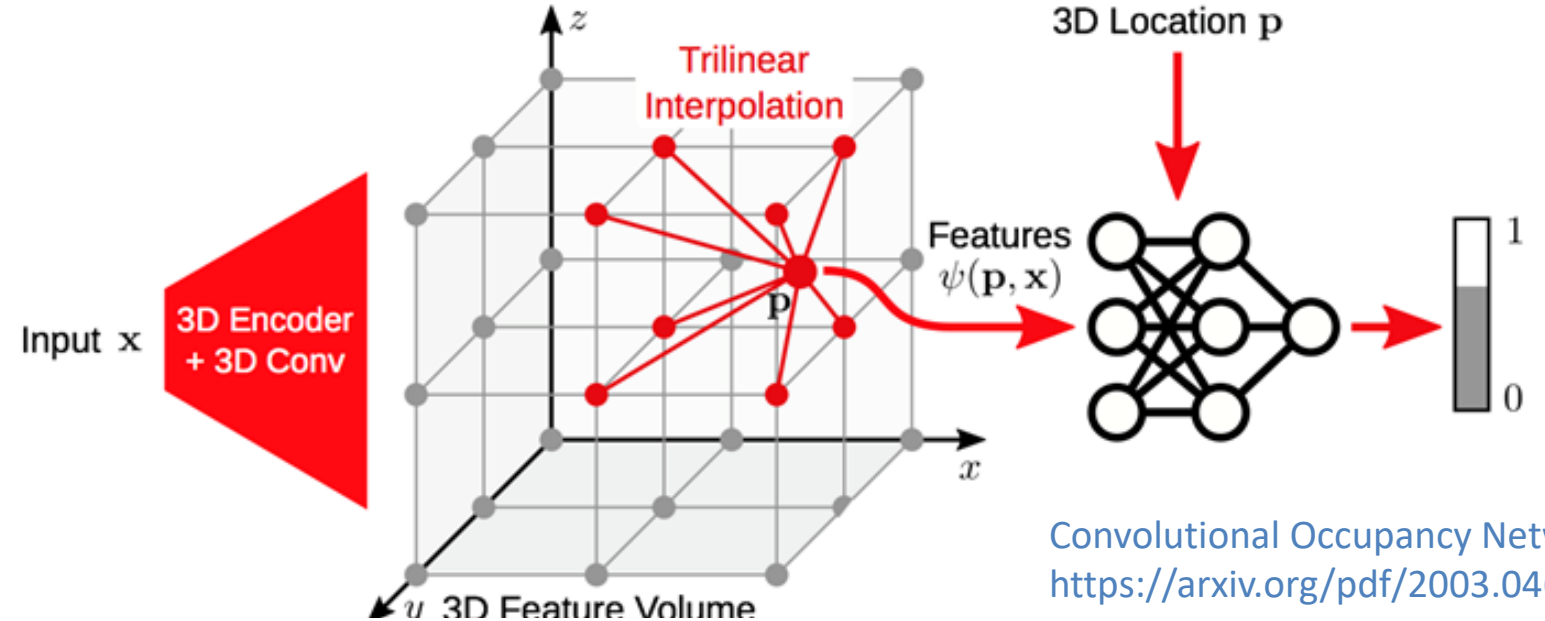
Single-Shape DeepSDF

MLS

Instead of mapping a 1D **global** vector to per-point implicit, map 3D **local** features to per-point implicit.

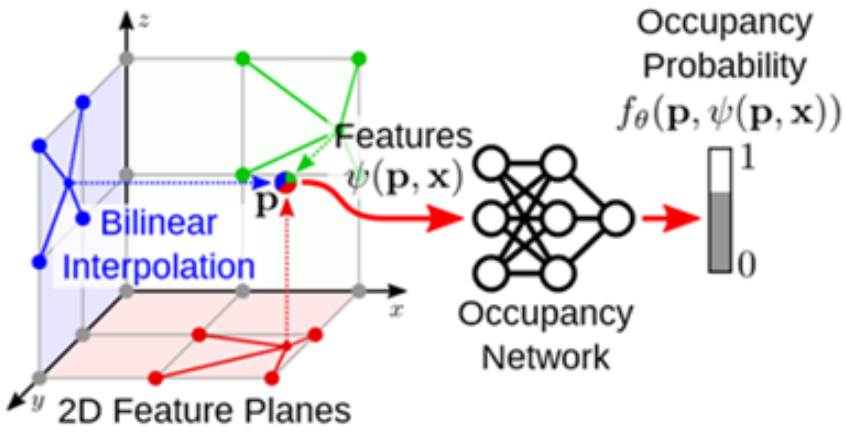


Convolutional Occupancy Networks

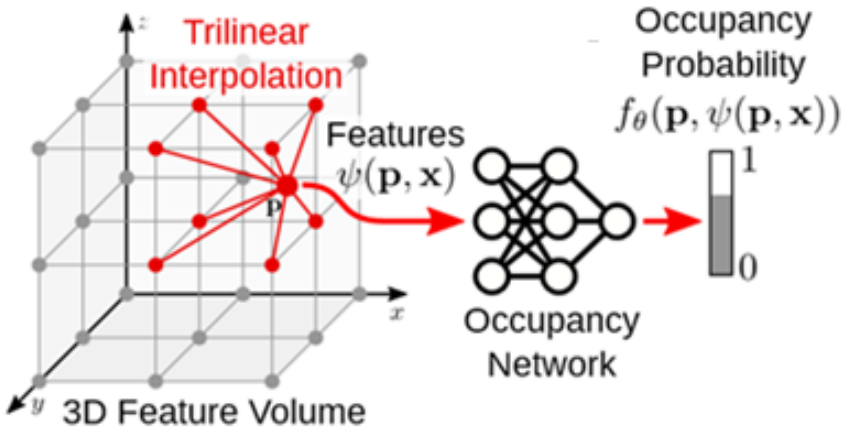


Convolutional Occupancy Networks, ECCV 2020
https://arxiv.org/pdf/2003.04618.pdf

Instead of mapping a 1D **global** vector to per-point implicit, map “multi-plane” **local** features to per-point implicit.

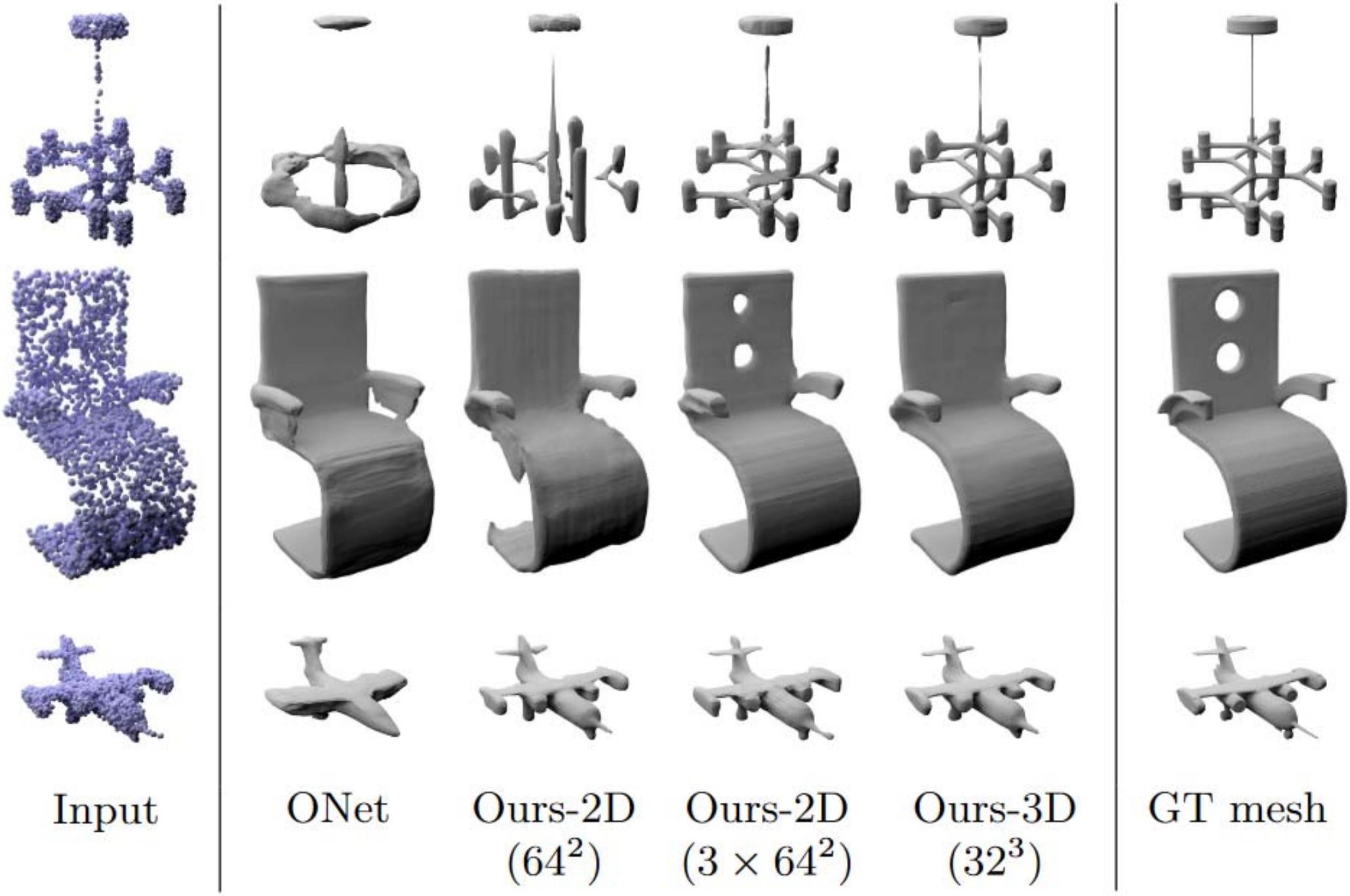


Convolutional Multi-Plane Decoder



Convolutional Volume Decoder

Convolutional Occupancy Networks, ECCV 2020
https://arxiv.org/pdf/2003.04618.pdf

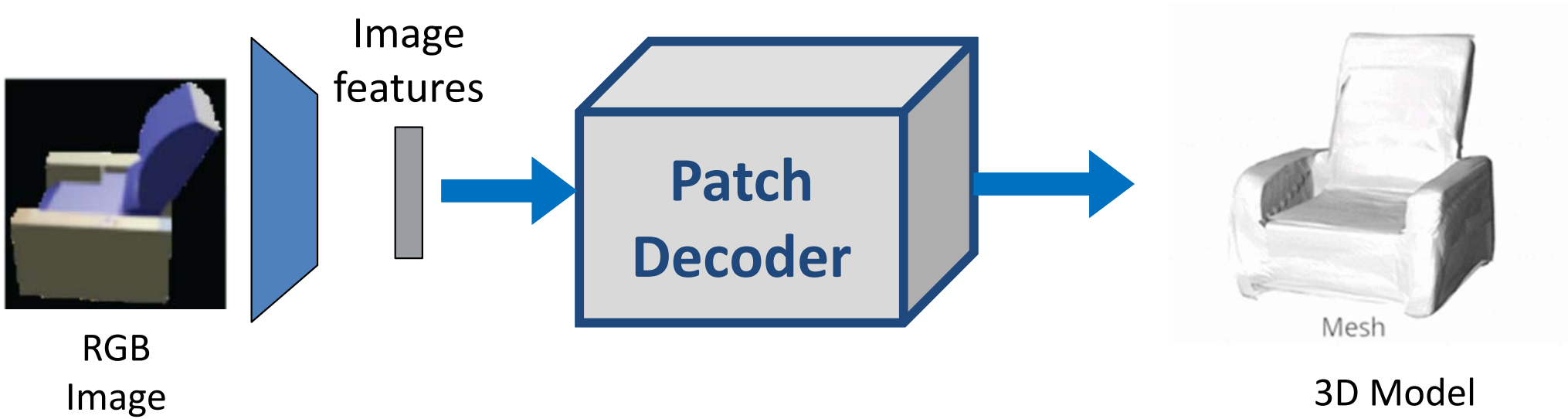


Convolutional Occupancy Networks, ECCV 2020
https://arxiv.org/pdf/2003.04618.pdf

How to generate shapes/scenes?

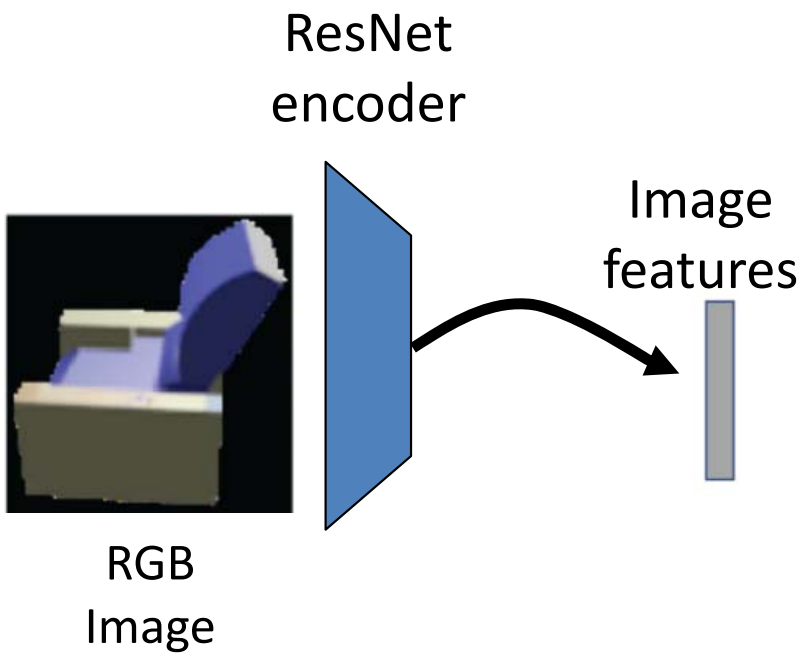
- **Encoder-Decoders**
  - Case Study: Multi-view decoder
  - Case Study: Implicit decoder
  - **Case Study: Patch decoder**
  - Case Study: Mesh Decoder
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

Image-to-patches



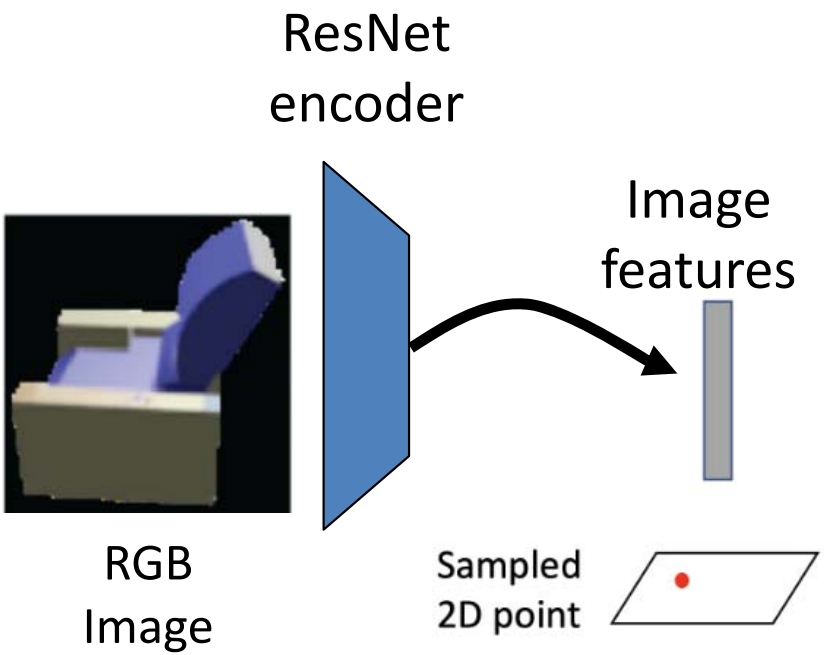
AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

Image-to-patches



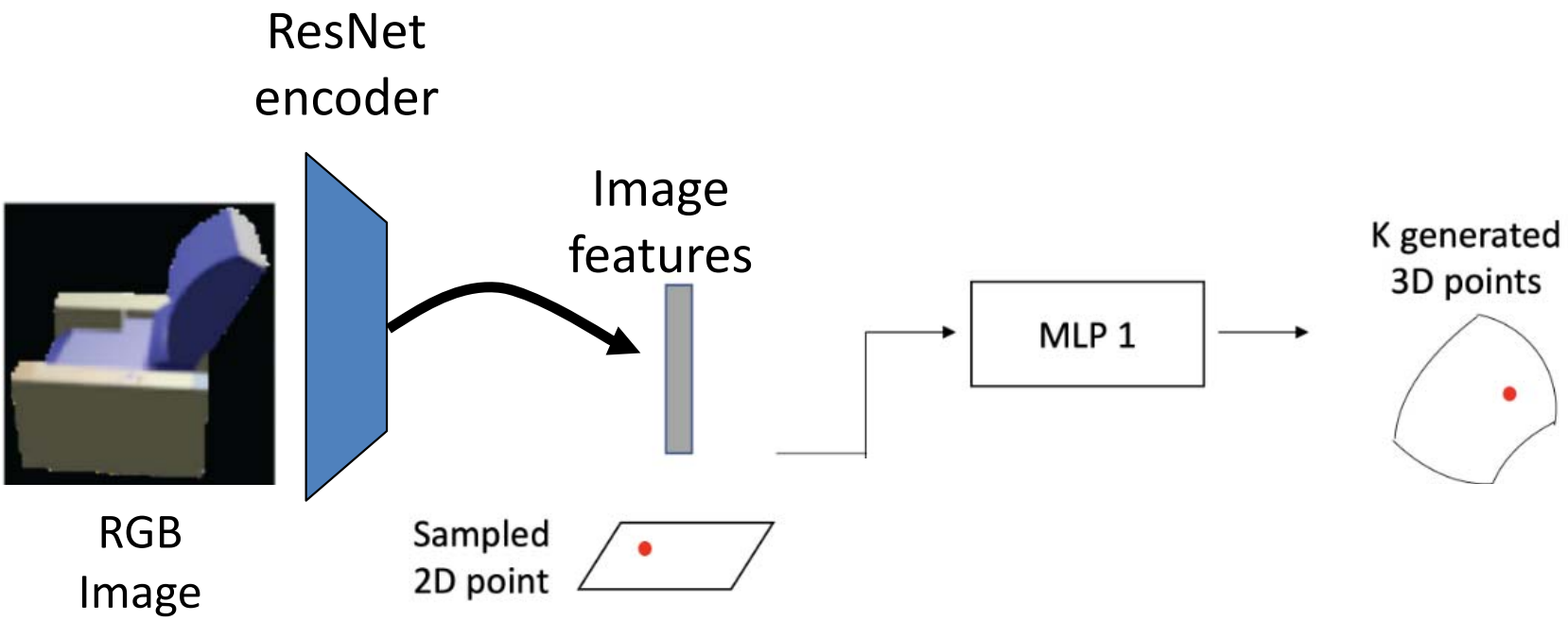
AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

Image-to-patches



AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

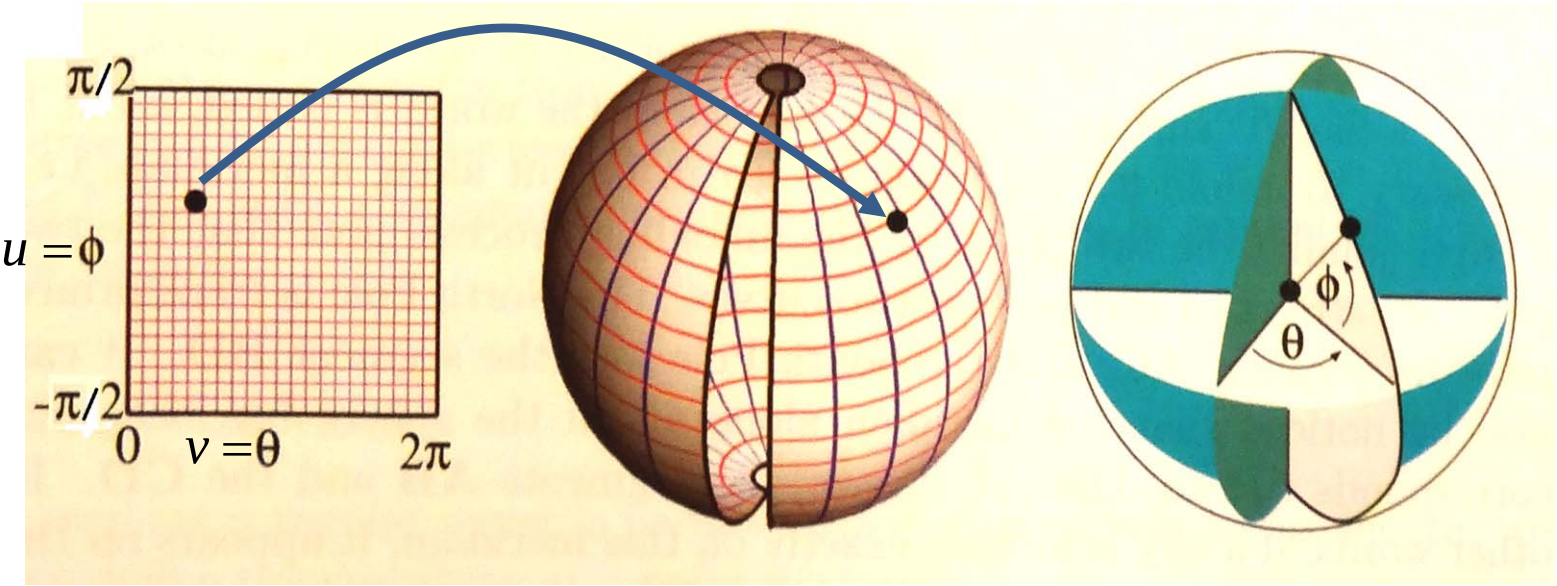
Image-to-patches



AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

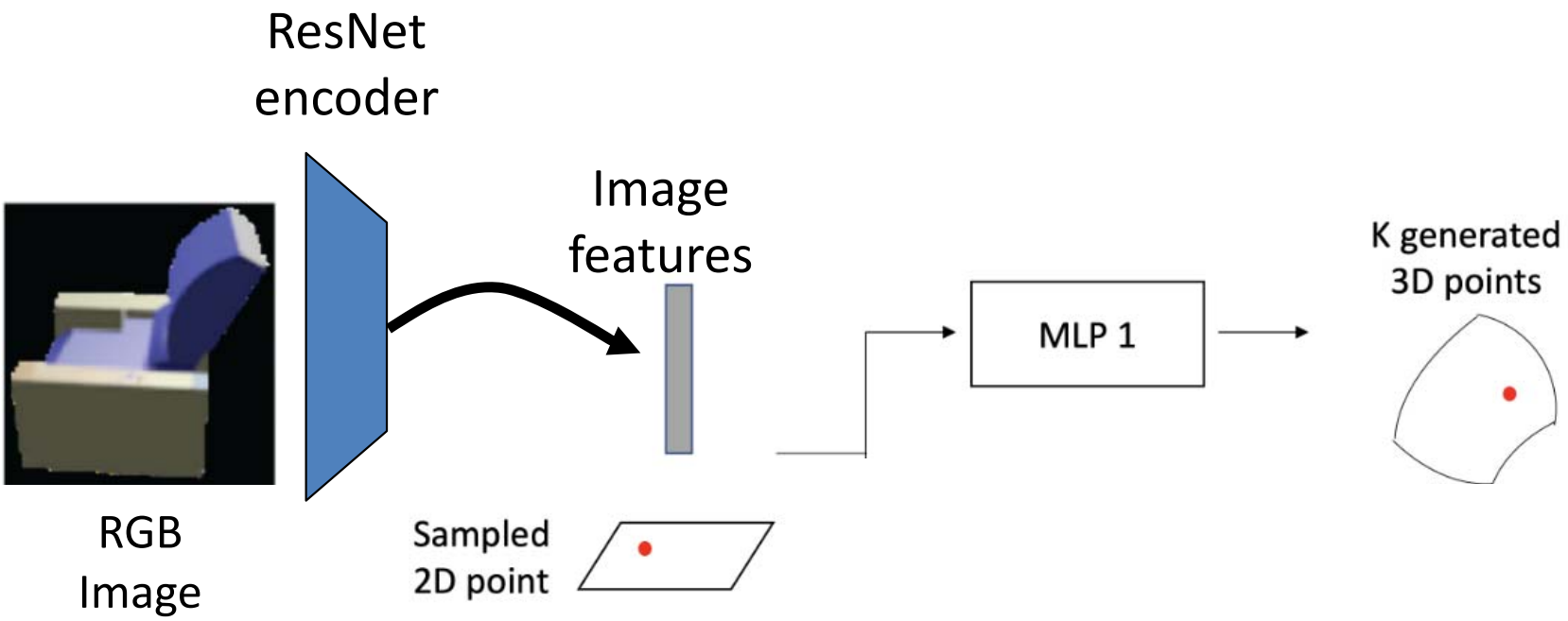
Why patches?

A (smooth) surface in 3D can be thought of as a map from a 2D set of parameters to 3D space



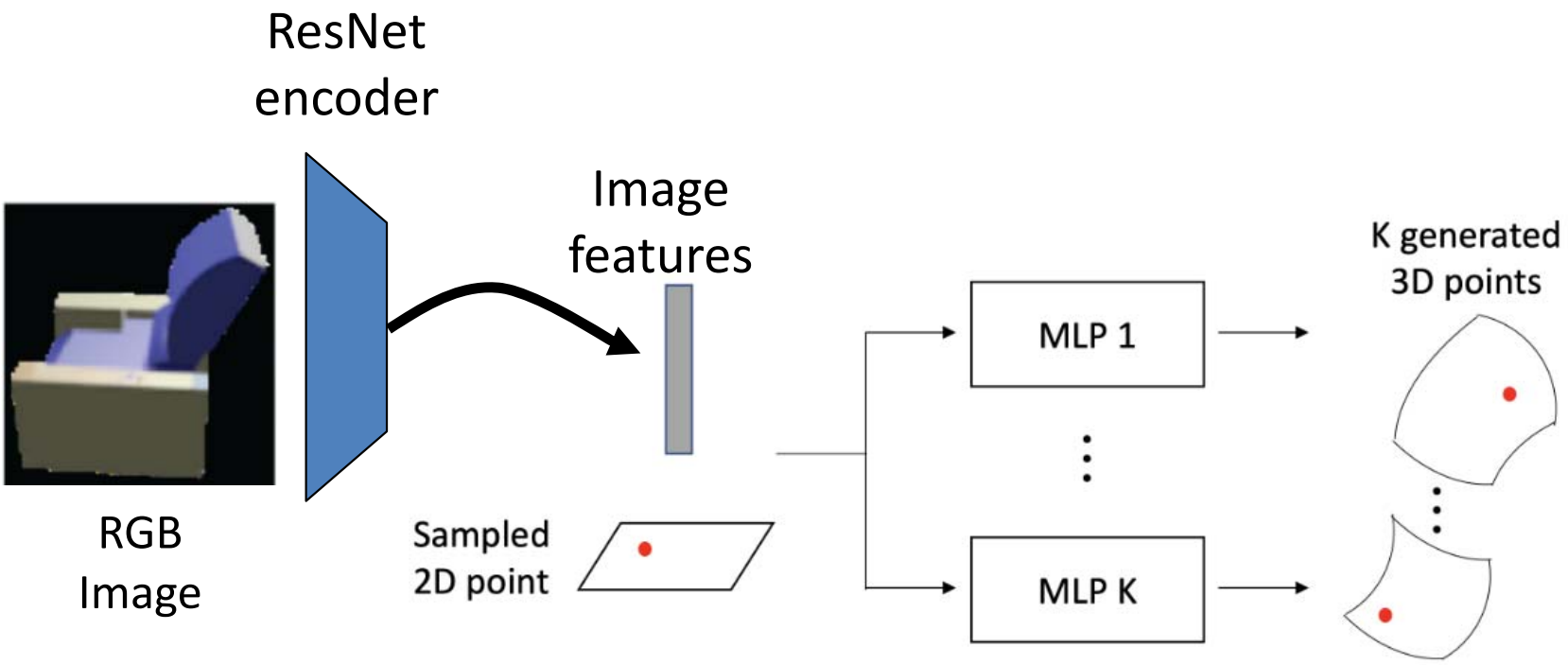
$$x(u,v) = R\cos u \sin v$$
$$y(u,v) = R\sin u \sin v$$
$$z(u,v) = R\cos v$$
$$u \in (0, 2\pi), v \in (-\pi/2, \pi/2)$$

Image-to-patches



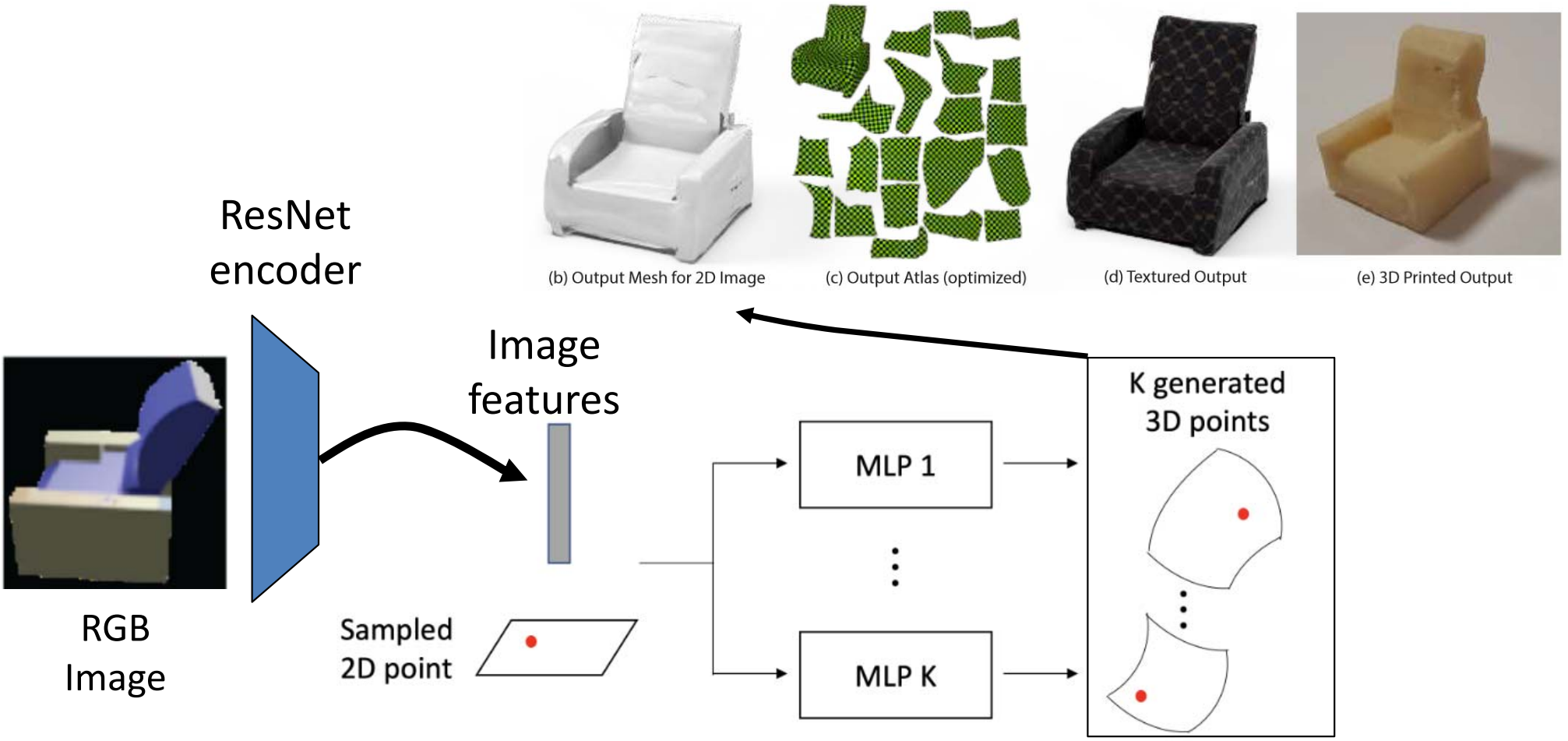
AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

Image-to-patches



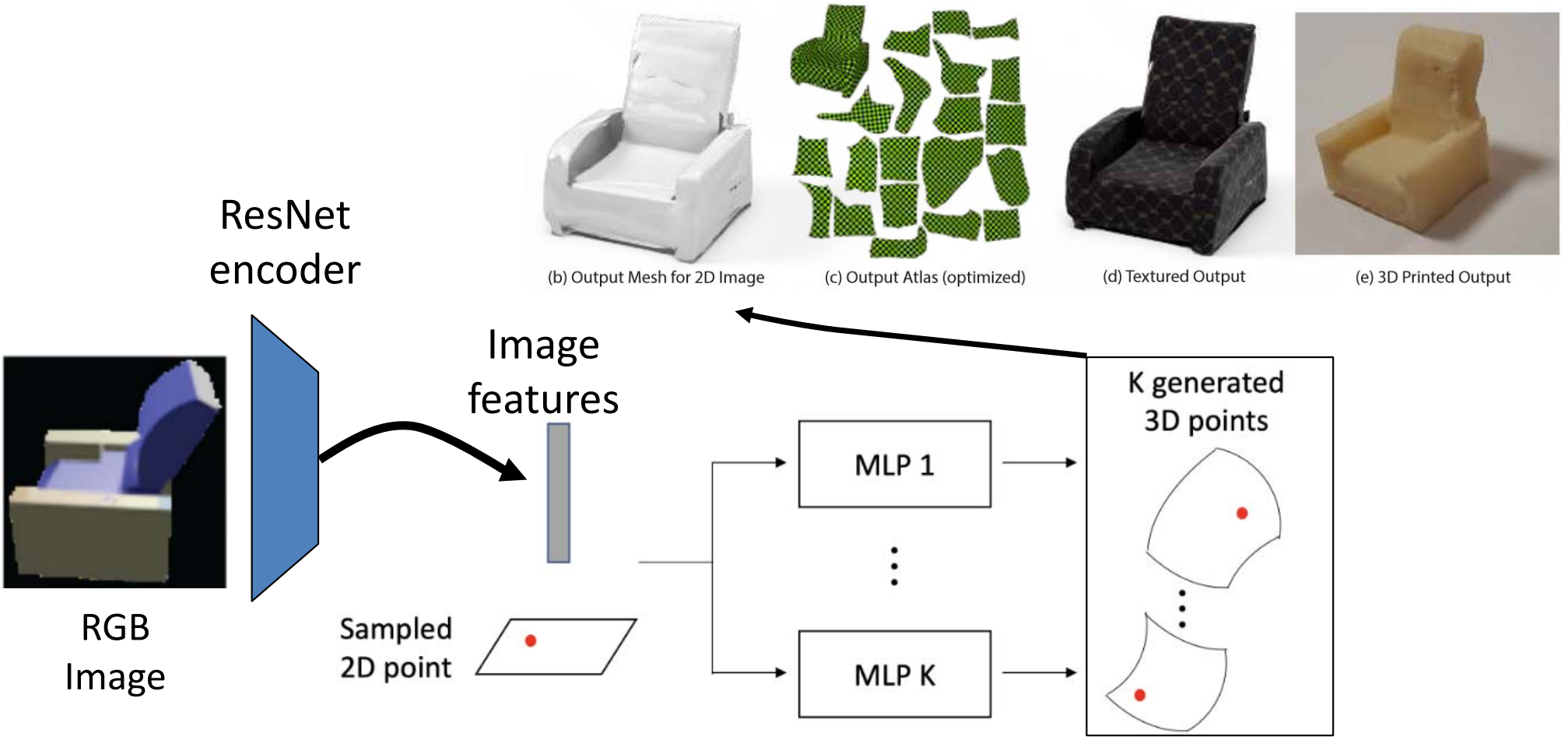
AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

Image-to-patches



AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation, Groueix et al. 2018

Image-to-patches



Given a ground-truth set $\boldsymbol{T}$ of points (samples from a training mesh) and the set $\boldsymbol{G}$ of generated points from all patches, use the **Chamfer distance** as loss:

$$d(\mathbf{T},\mathbf{G}) = \sum_{\mathbf{t}\in\mathbf{T}} \min_{\mathbf{g}\in\mathbf{G}} \|\mathbf{t}-\mathbf{g}\| + \sum_{\mathbf{g}\in G} \min_{\mathbf{t}\in\mathbf{T}} \|\mathbf{t}-\mathbf{g}\|$$

Results

RGB
Image input

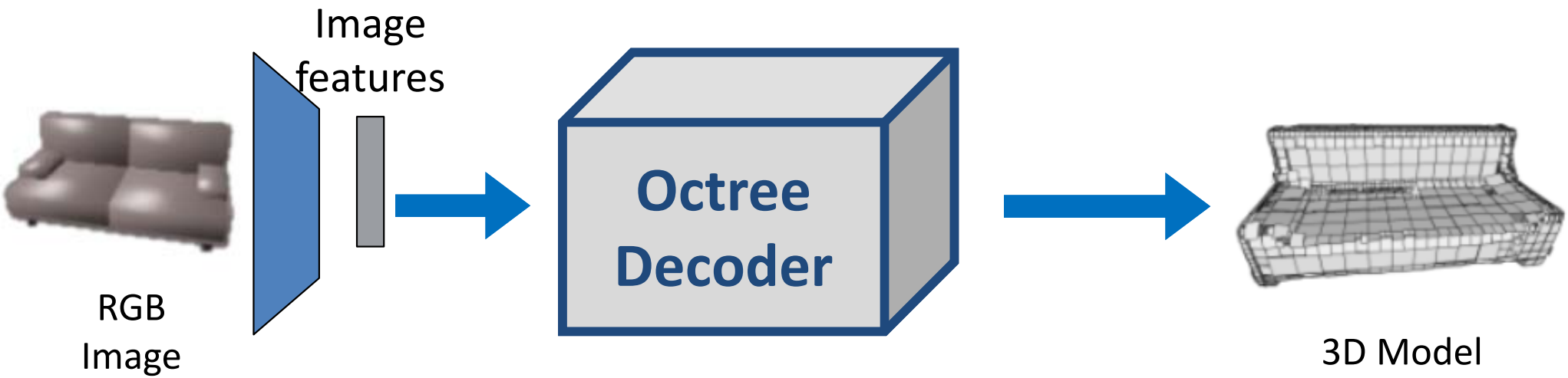
Voxel Network
(3DR2N2)

AtlasNet

How to generate shapes/scenes?

- **Encoder-Decoders**
  - Case Study: Multi-view decoder
  - Case Study: Implicit decoder
  - **Case Study: Patch decoder *within an octree***
  - Case Study: Mesh Decoder
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

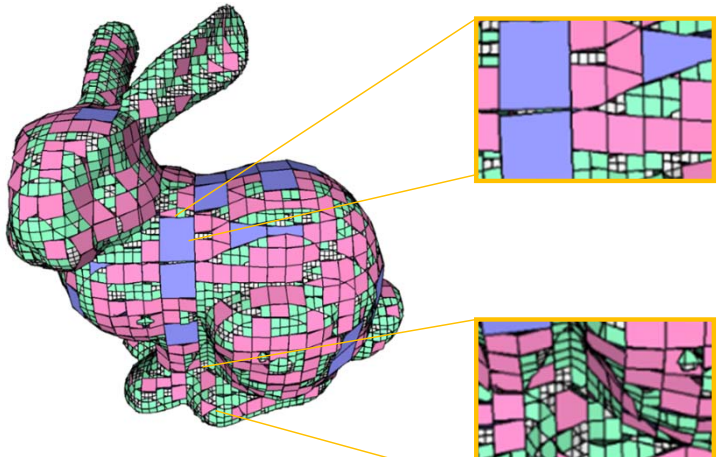
Image-to-patches



Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN: key idea

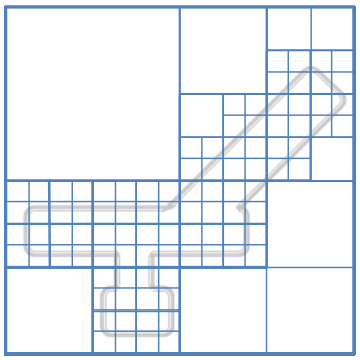
Represent the input shape with adaptive planar patches



Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN: key idea

Patches are created for each cell in a **generated octree**

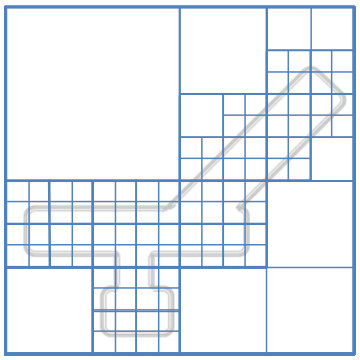


Octree:
subdivide if non-empty

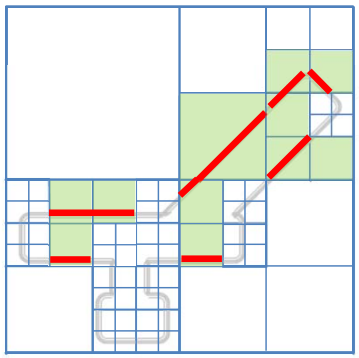
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN: key idea

Patches are created for each cell in a **generated octree**



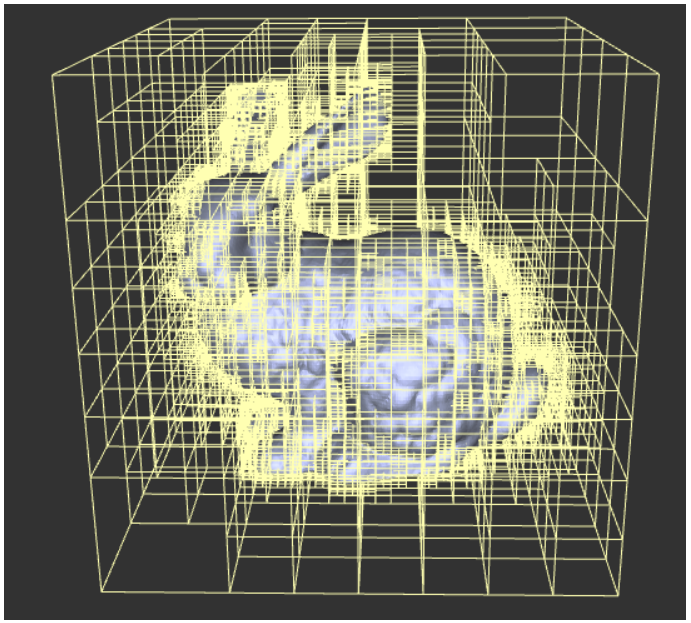
Octree:
subdivide if non-empty

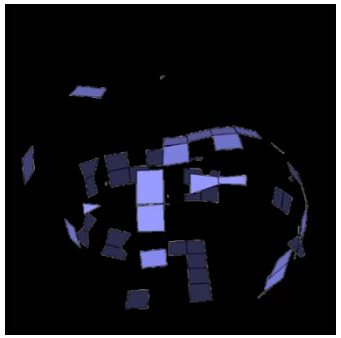


Adaptive octree:
subdivide if non-empty &&
surface-poorly-approximated

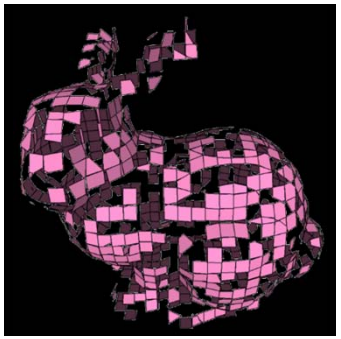
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Patch-Based Adaptive Octree

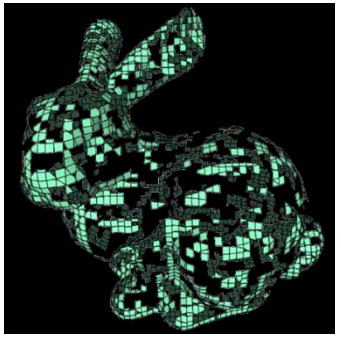




Planar patches at 4-level



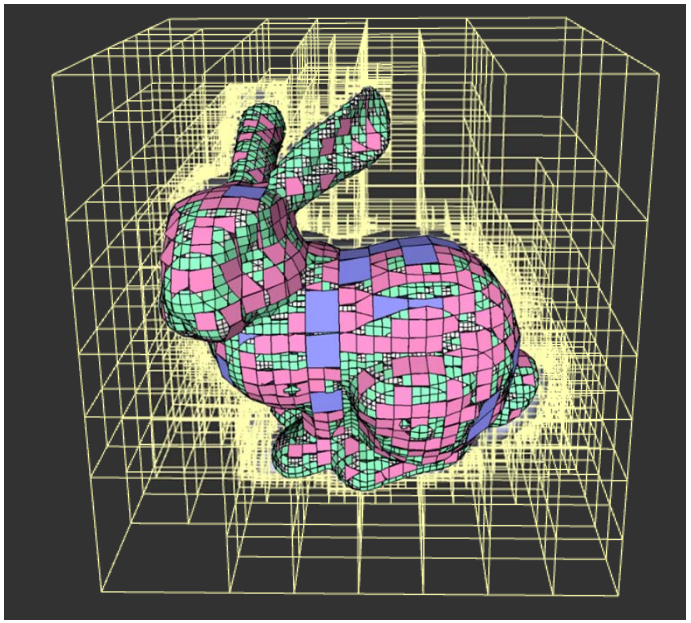
Planar patches at 5-level



Planar patches at 6-level

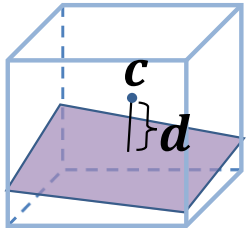
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

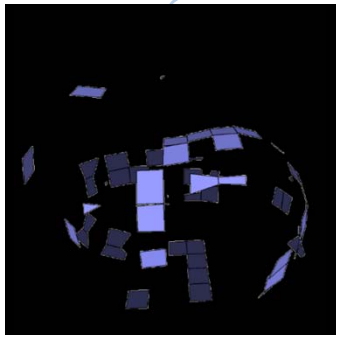
Patch-Based Adaptive Octree



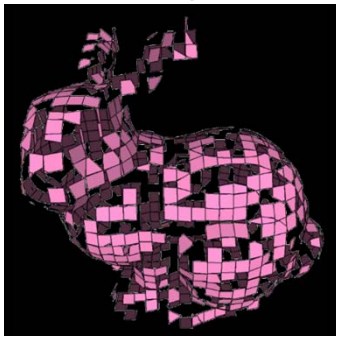
Planar patches

$$\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{c}) + d = 0$$

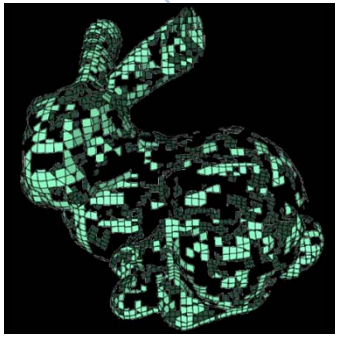




Planar patches at 4-level



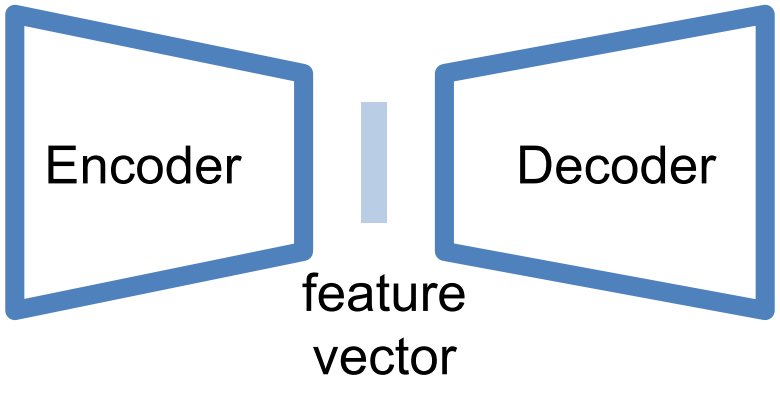
Planar patches at 5-level

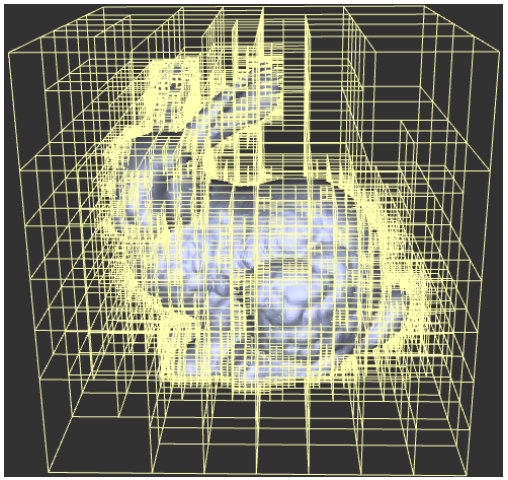


Planar patches at 6-level

Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Encoder and Decoder

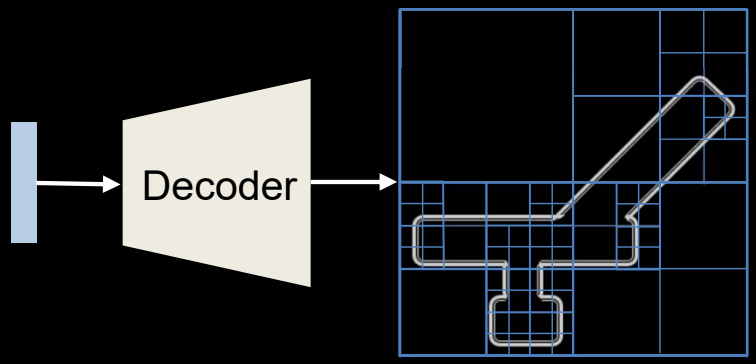




Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

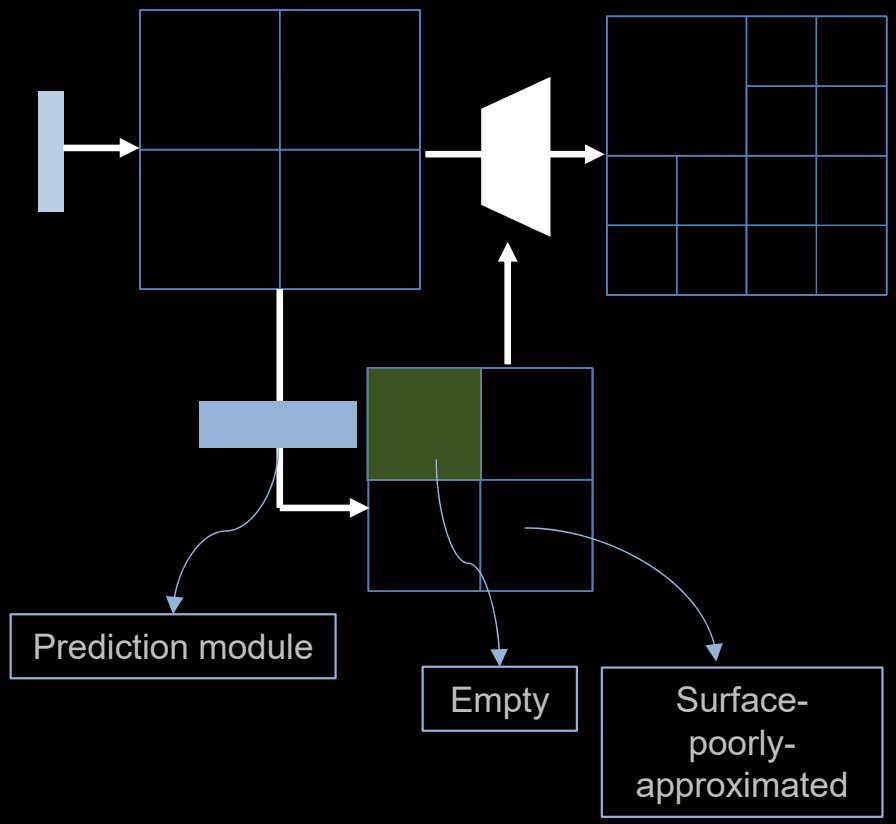
Adaptive O-CNN Decoder

How to generate unknown octree structure?



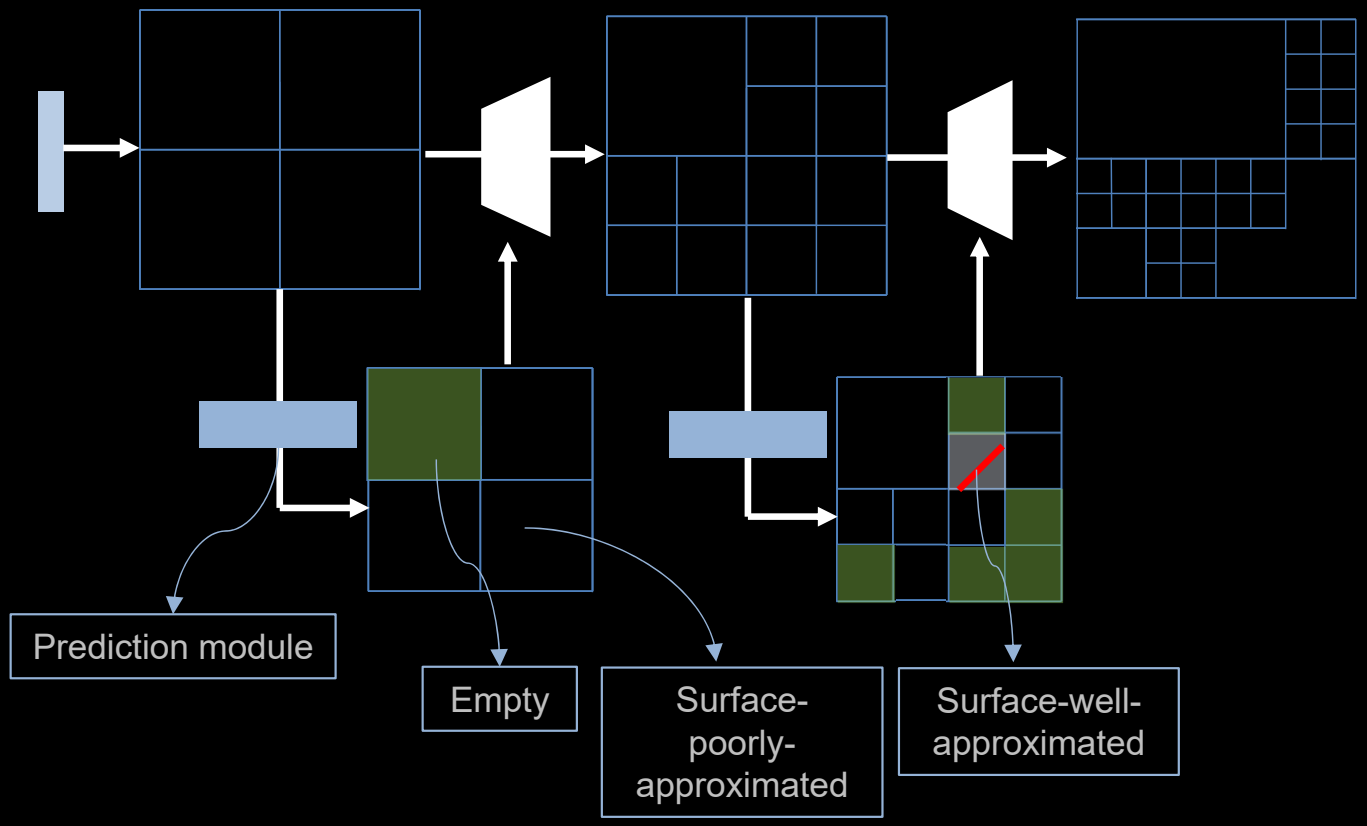
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN Decoder



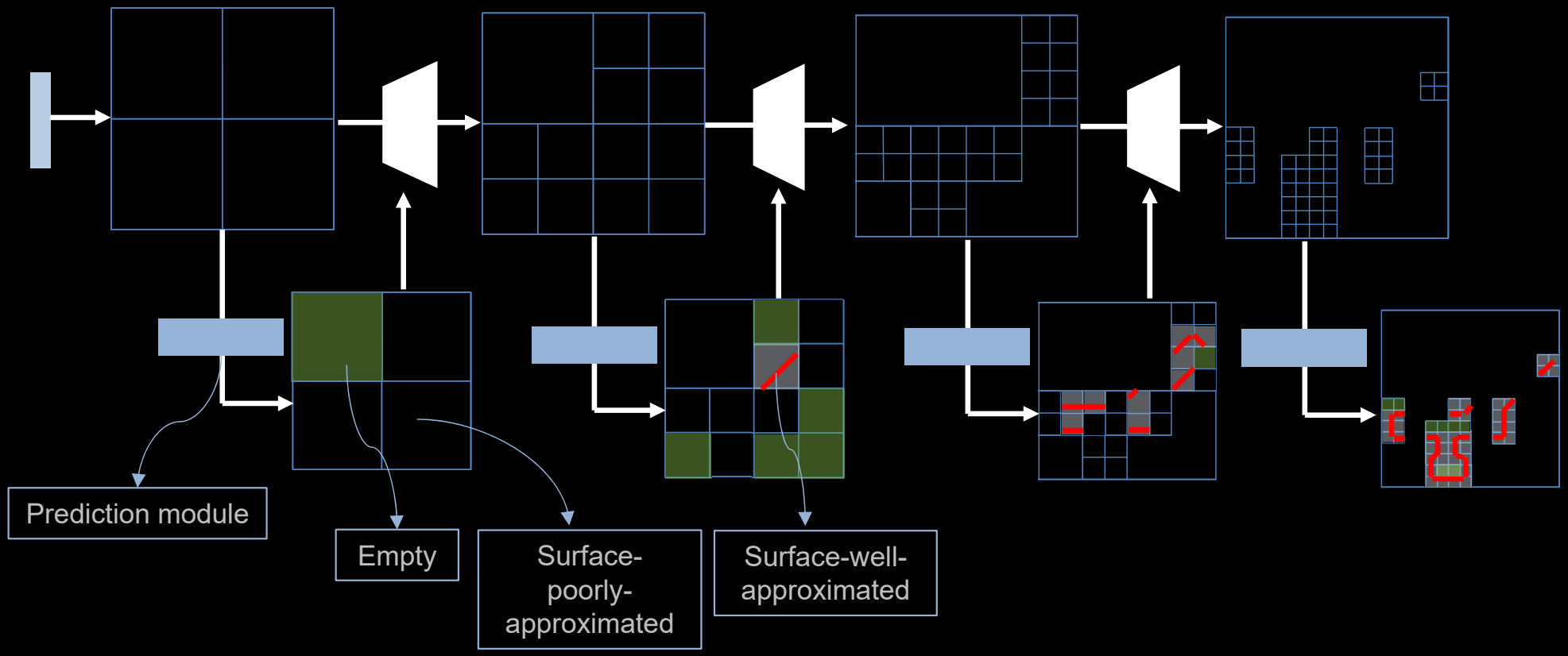
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN Decoder



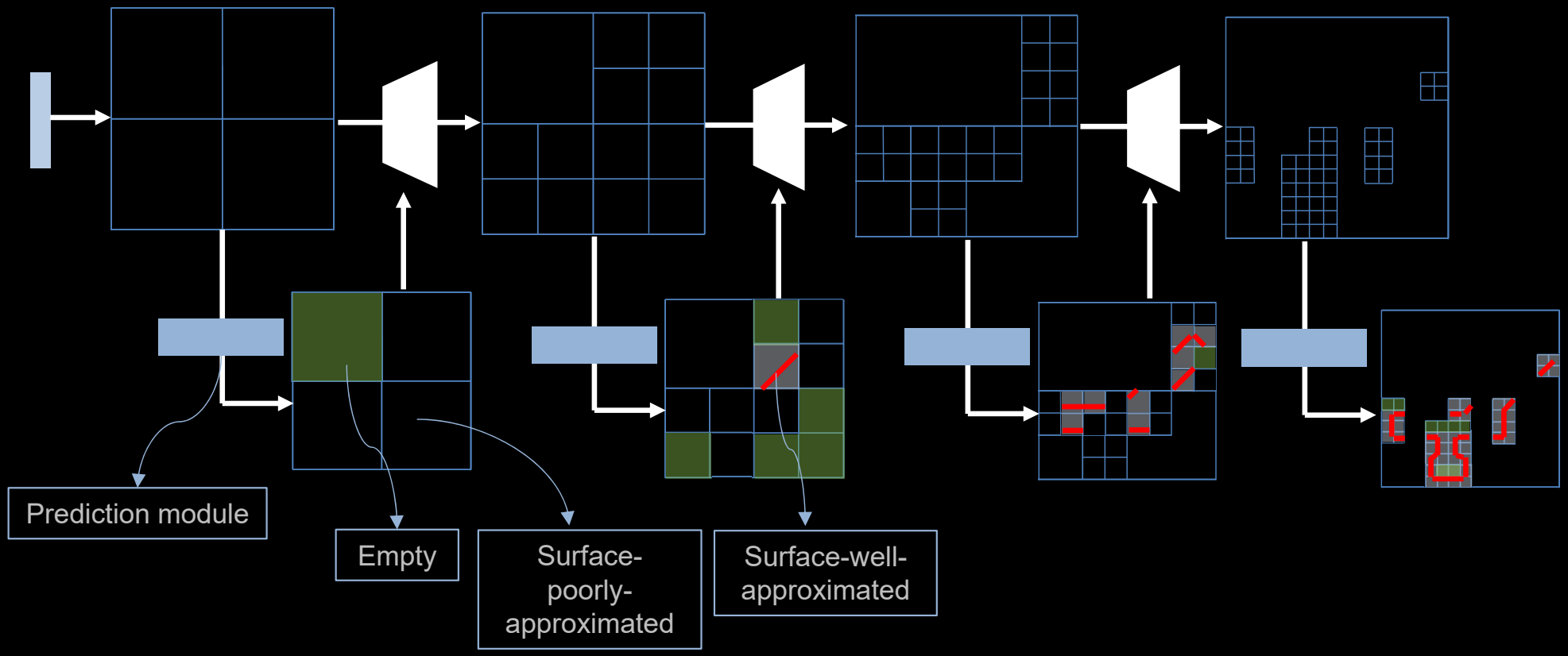
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN Decoder



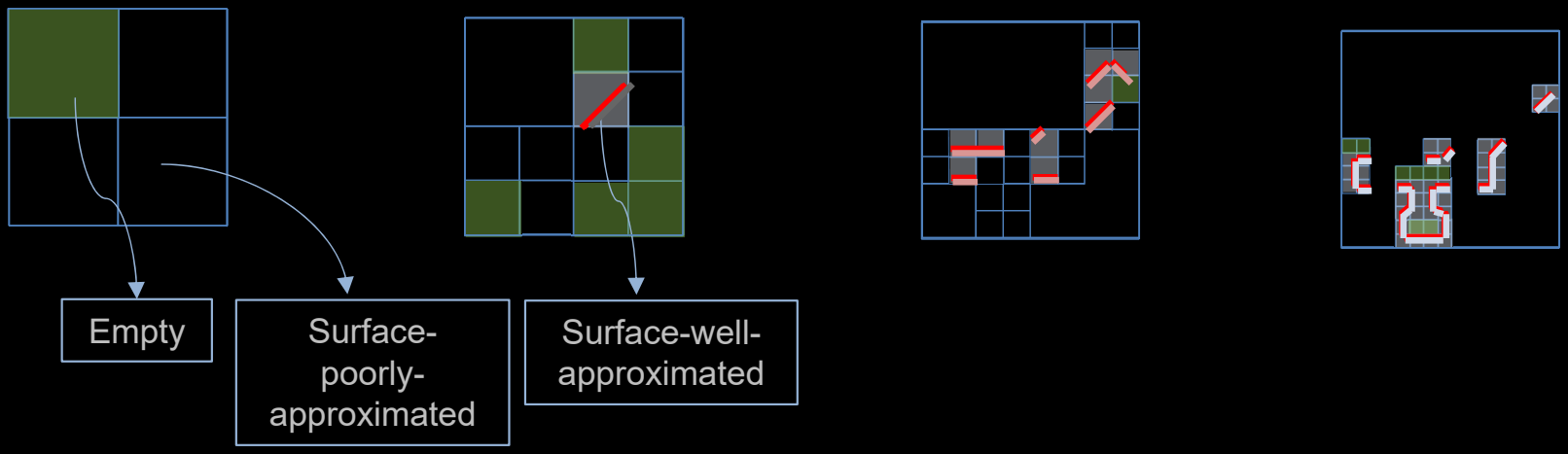
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN Decoder



Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

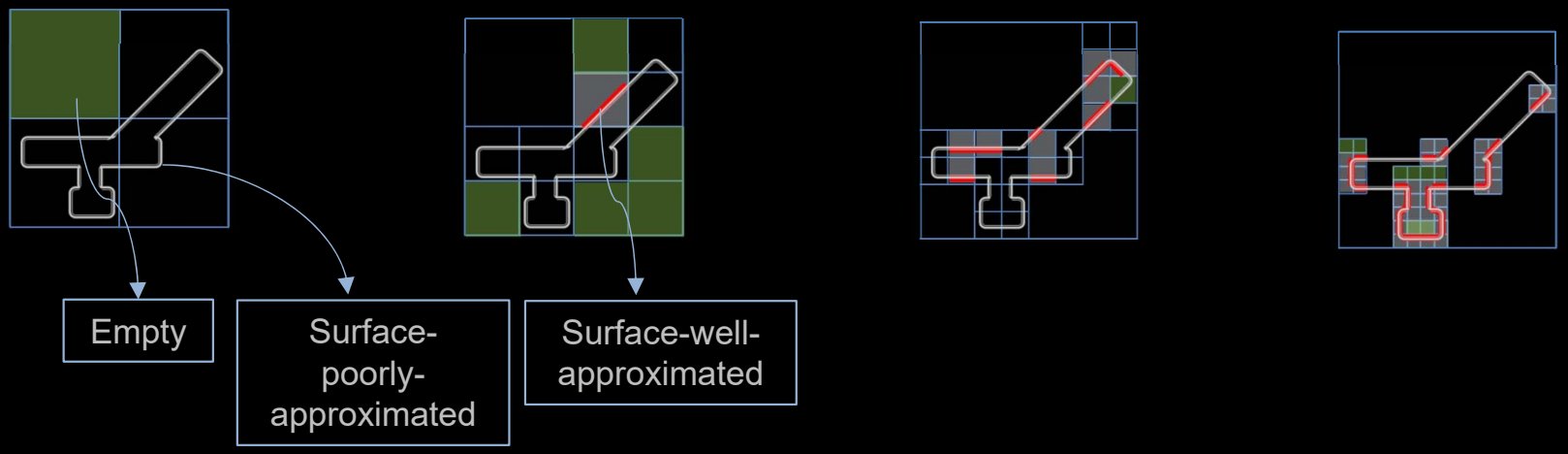
Adaptive O-CNN Decoder



Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Adaptive O-CNN Decoder: Loss Function

Combine cross-entropy for classifying cells and regression for plane parameters



Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

Results

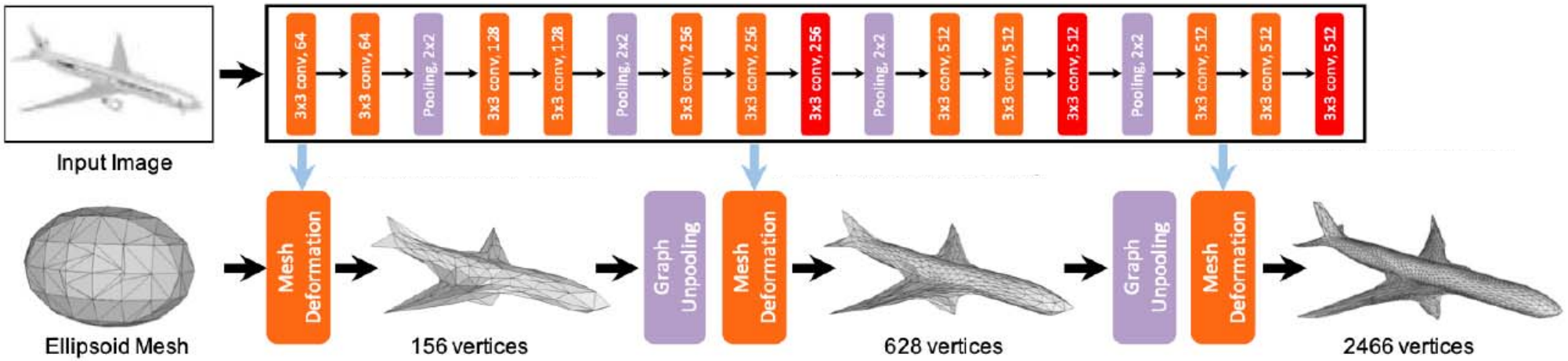
Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes, Wang et al. 2018

How to generate shapes/scenes?

- **Encoder-Decoders**
  - Case Study: Multi-view decoder
  - Case Study: Implicit decoder
  - Case Study: Patch decoder
  - **Case Study: Mesh Decoder**
- Generative Adversarial Networks
- Variational Autoencoders
- Autoregressive Models
- Diffusion Models

Pixel2Mesh

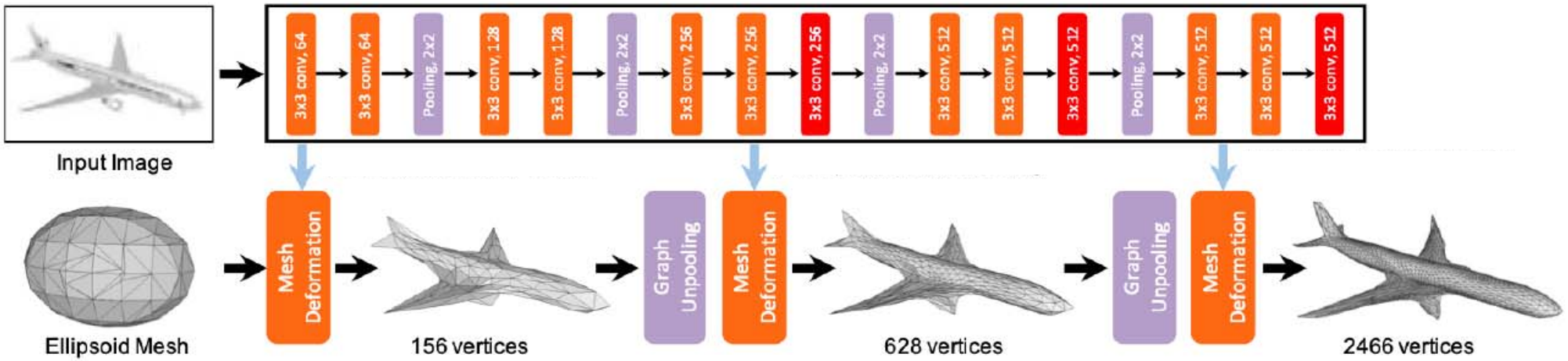
Learns to deform a template mesh guided by an input image.

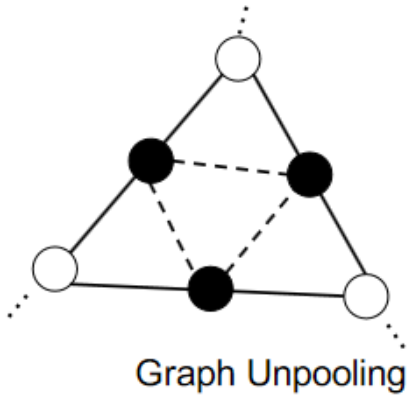


Pixel2Mesh, Wang et al. 2018

Pixel2Mesh

Learns to deform a template mesh guided by an input image.
The decoder predicts vertex positions via graph conv+unpool.





Pixel2Mesh, Wang et al. 2018

Results

Pixel2Mesh, Wang et al. 2018

See also "**Deep Mesh Reconstruction from Single RGB Images via Topology Modification Networks", Pan et al., 2019** that tries to predict which triangles to remove to replicate holes