REAACH-ID

Online Symposium – 13-14 Ottobre 2020

CompoNET: a geometric deep learning approach in Architecture

Alberto Tono, Meher Shashwat Nigam, Cecilia Bolognesi, Amirhossein Ahmadnia

Computational Design Institute, IIIT Hyderabad, Politecnico di Milano,

Alberto.tono@cd.institute, meher.shashwat@students.iiit.ac.in,

cecilia.bolognesi@polimi.it , amirhossein.ahmadnia@mail.polimi.it

POLITECNICO
MILANO 1863

Agenda

1

2

1. Use Cases: Picture2BIM
2. Approach
3. Related Work

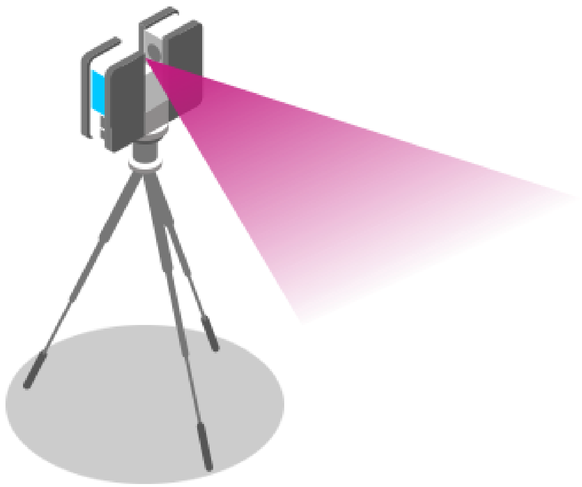
3

4. Solution
5. Future Research

4

5

Use Cases



Scan Building

Expensive Process

Expensive Equipment

Several Man Hours

Level of Detail

Point Clouds

RGBD Pictures

Accessible

R&D required

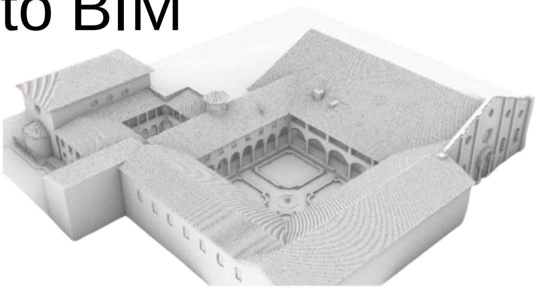
Granular Level of Detail

High-Resolution Images

Use Cases

Augmented Reality Applications

Scan to BIM



Autonomous Driving: Urban Design

Robotics in the AEC

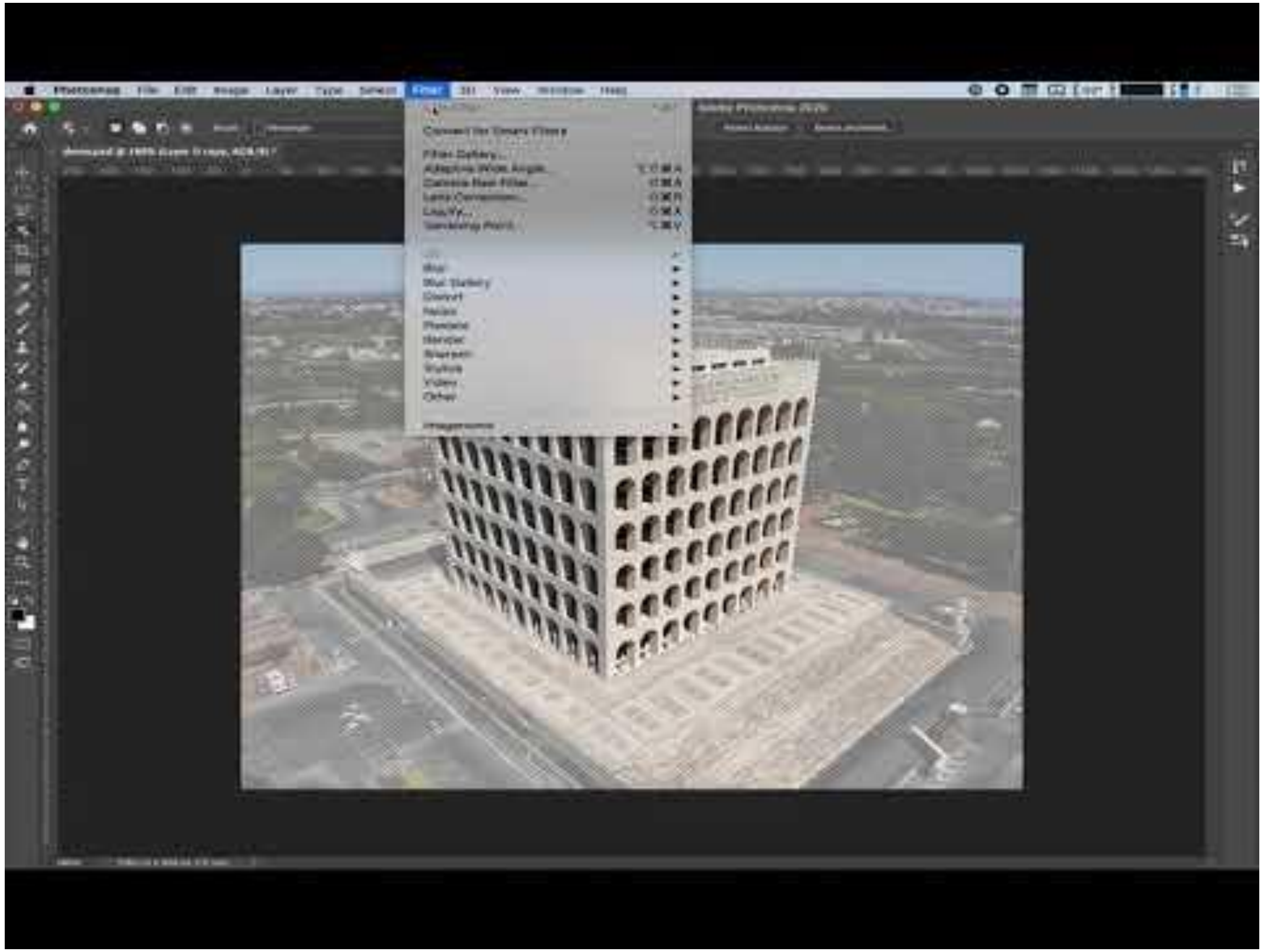
Aerospacial, Rover

Assembly

Surveys

Authoring 3D Content

Use Cases



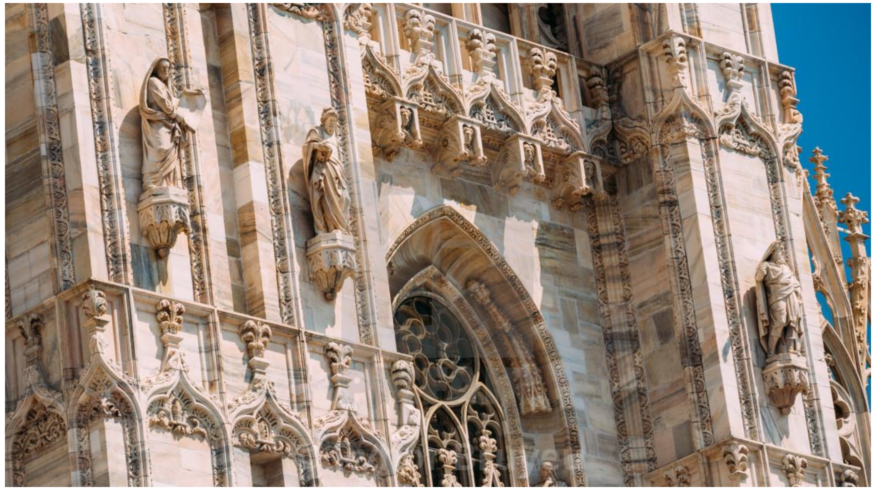
Learning to Infer and Execute 3D Shape Programs, Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, Jiajun Wu

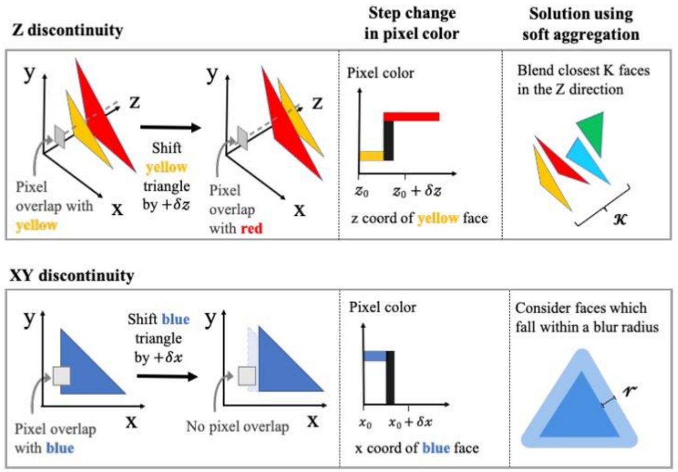
Use Cases

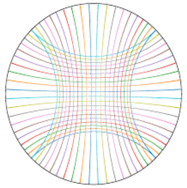
Approach: Differentiable Render

1

2







[Geoopt](#)
[Geomstats](#)
[Pymanopt](#)

kornia

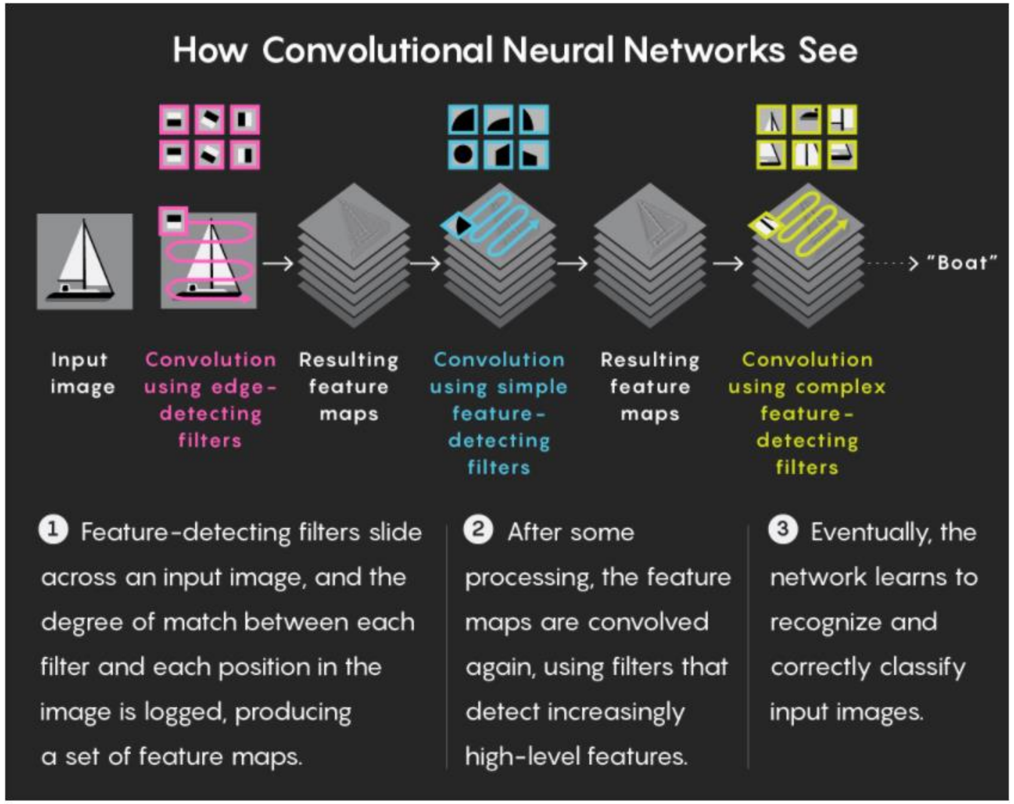
Kaolin

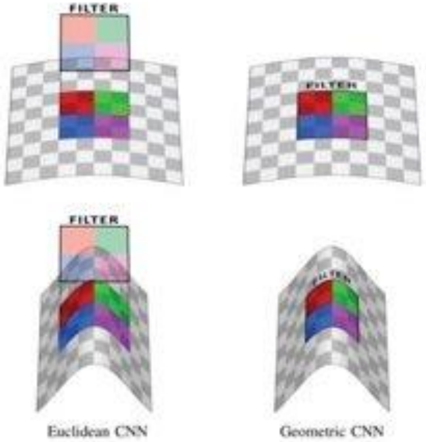
PyTorch geometric

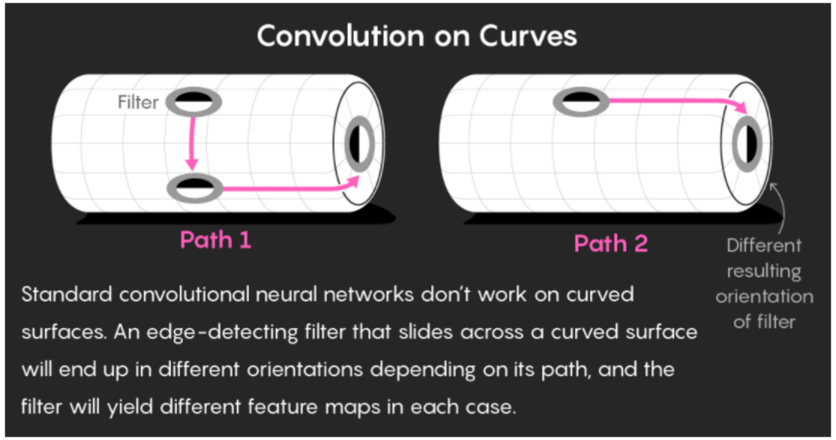
PyTorch3D

TensorFlow Graphics

Approach





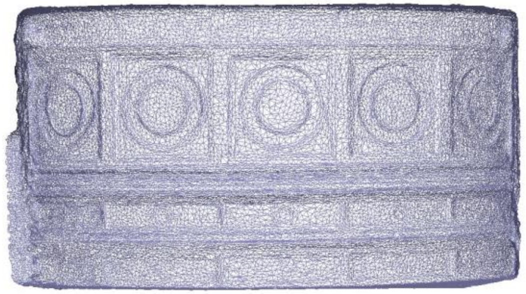


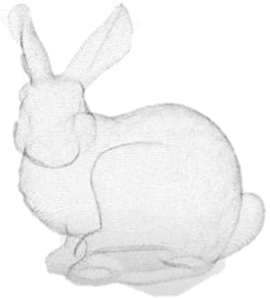
Lucy Reading-Ikkanda/Quanta Magazine

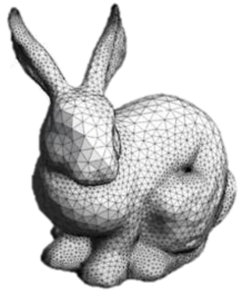
Datasets

1

2







Related Work

1

- Procedural Approach

2

- 3D-R2N2
- Pix2Mesh

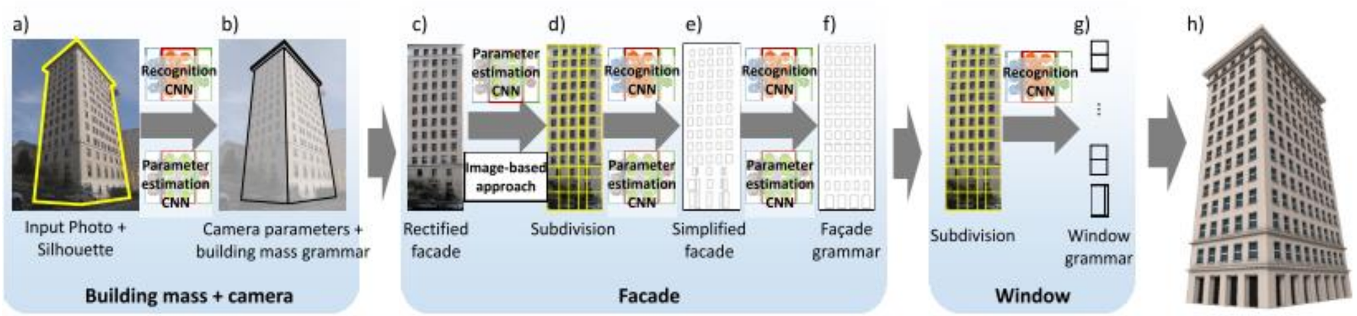
3

- Occupancy Network
- N3MR
- NeRF
- Mesh R-CNN
- 3D-Part Assembly

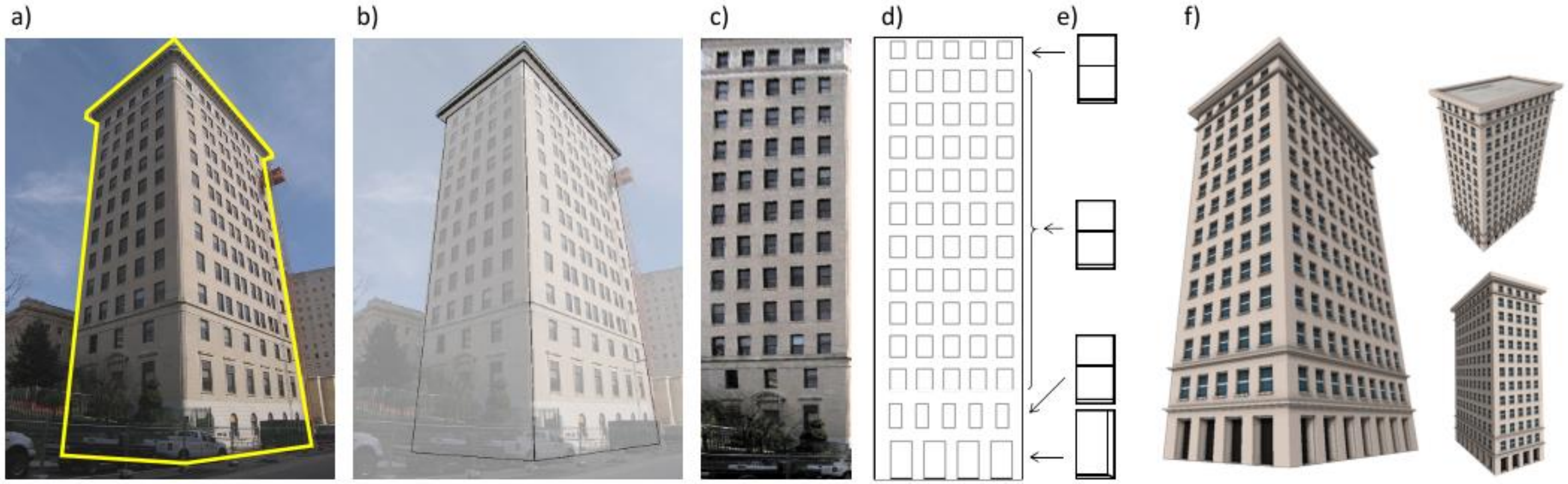
1

Procedural Modeling

2



3



DeepFacade: A Deep Learning Approach to Facade Parsing Hantang Liu , Jialiang Zhang , Jianke Zhu , Steven C.H. Hoi, Alibaba-Zhejiang

Procedural Modeling of a Building from a Single Image Gen Nishida, Adrien Bousseau, Daniel G. Aliaga

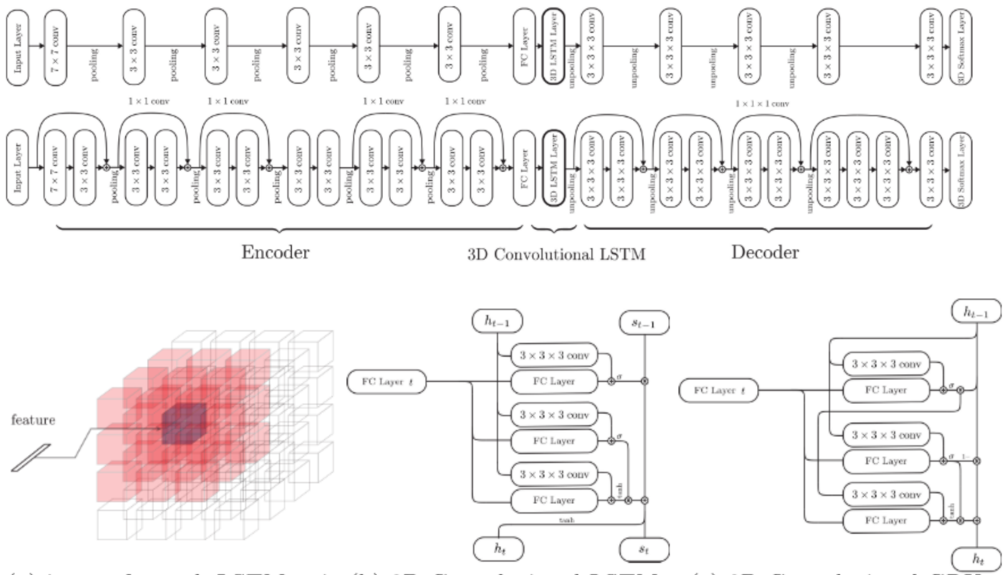
3D-R2N2

1

2

3

(a) Images of objects we wish to reconstruct (b) Overview of the network

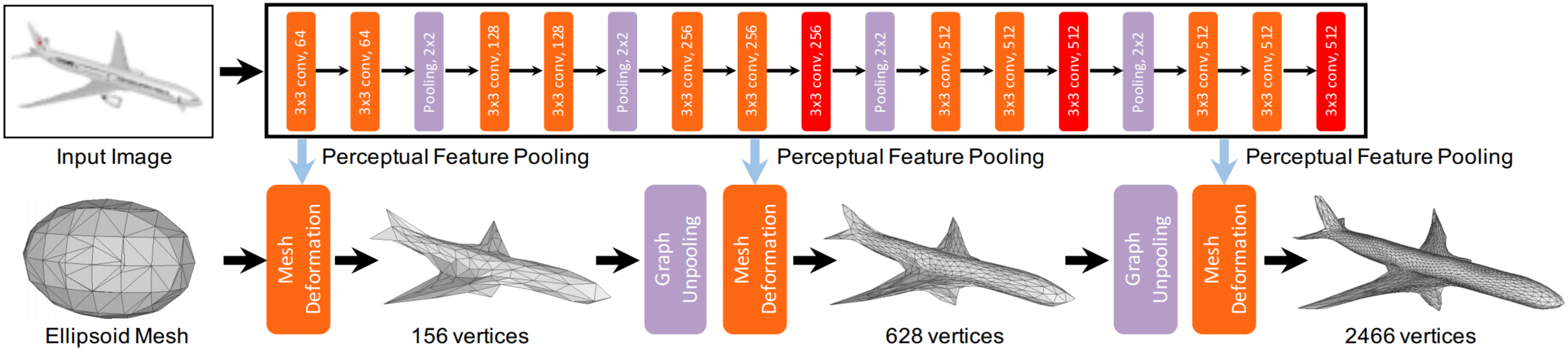


(a) inputs for each LSTM unit (b) 3D Convolutional LSTMs (c) 3D Convolutional GRUs

3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction Christopher B. Choy Danfei Xu, JunYoung Gwak, Kevin Chen Silvio Savarese

1 Pix2Mesh

2



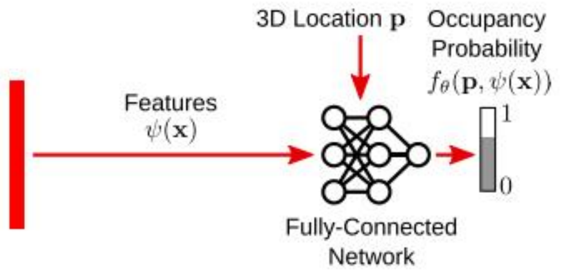
Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Li, Yu-Gang Jiang

Occupancy Network

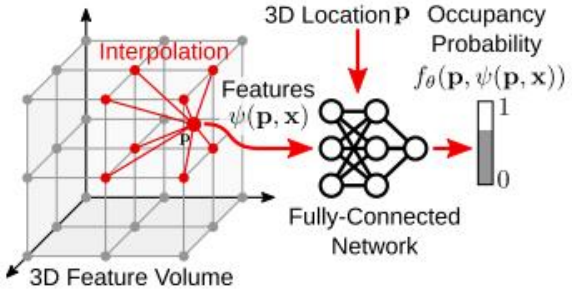
1

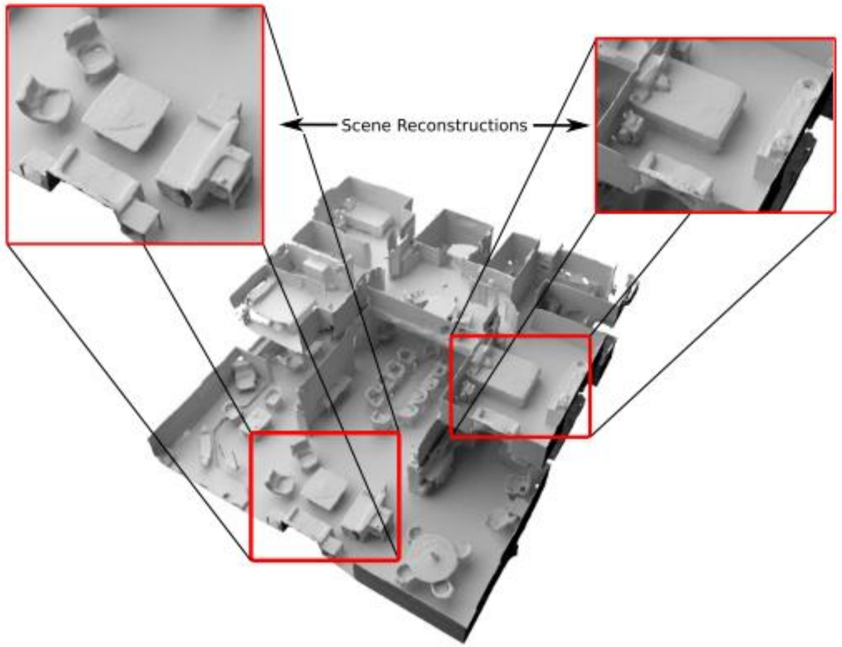
2

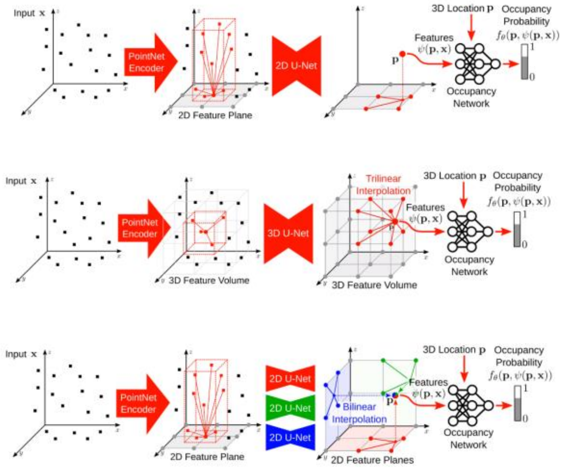
3

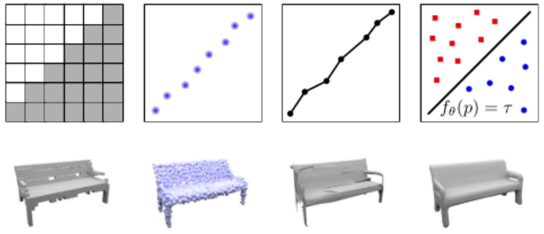


(a) Occupancy Network [26]









(a) Voxel (b) Point (c) Mesh (d) Ours

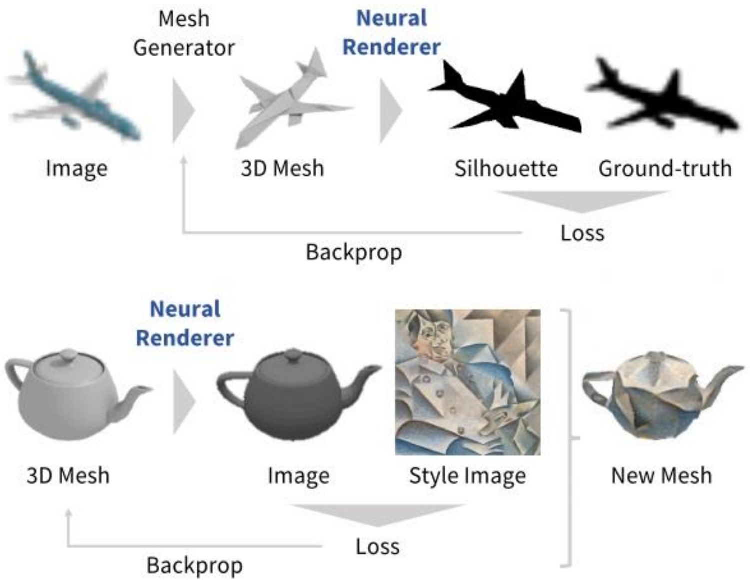
Convolution Occupacy Network: Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, Andreas Geiger

Occupancy Networks: Learning 3D Reconstruction in Function Space Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger

N3MR

1

2



3

Figure 1. Pipelines for single-image 3D mesh reconstruction (upper) and 2D-to-3D style transfer (lower).

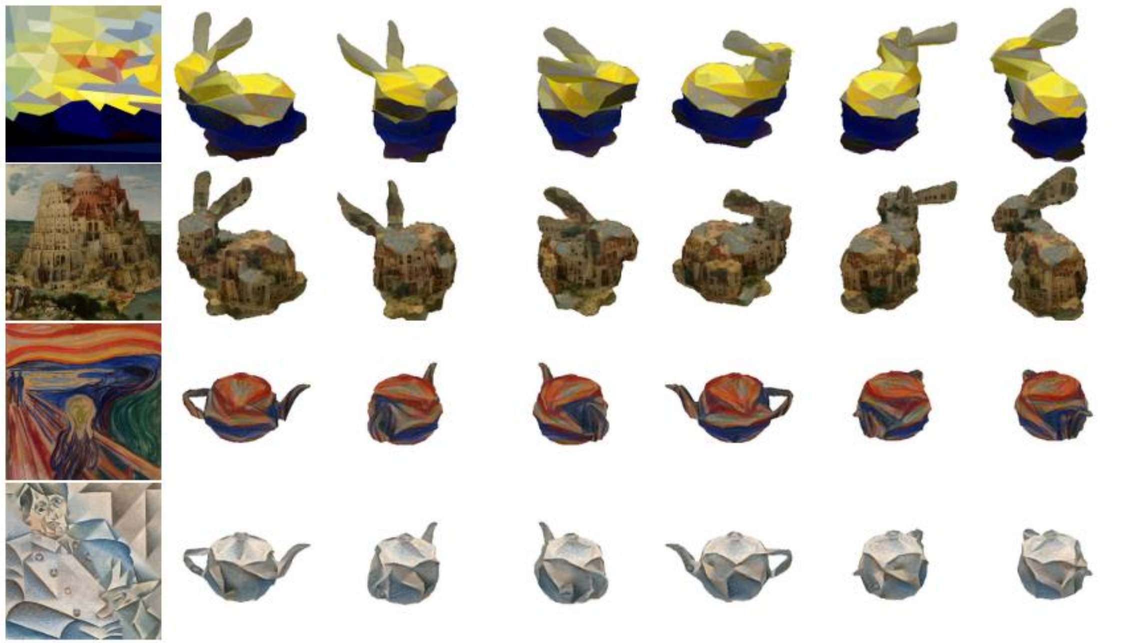


Figure 7. 2D-to-3D style transfer. The leftmost images represent styles. The style images are *Thomson No. 5 (Yellow Sunset)* (D. Coupland, 2011), *The Tower of Babel* (P. Bruegel the Elder, 1563), *The Scream* (E. Munch, 1910), and *Portrait of Pablo Picasso* (J. Gris, 1912).

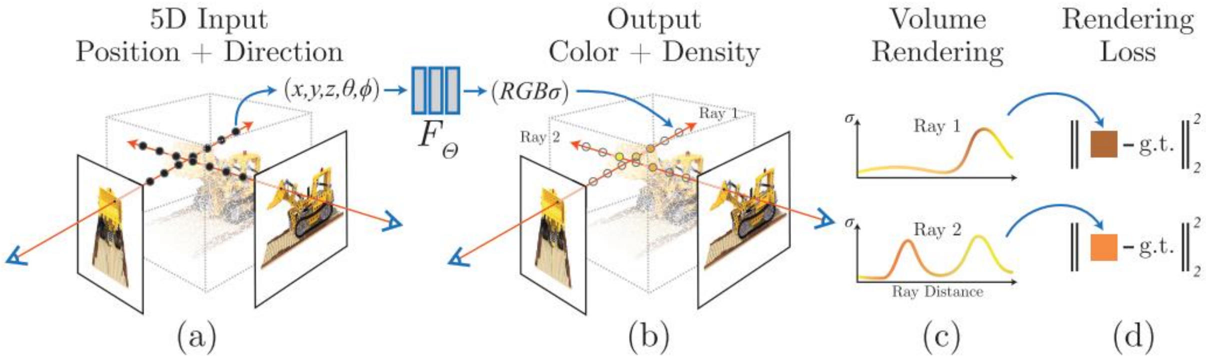
Neural 3D Mesh Renderer, Hiroharu Kato, Yoshitaka Ushiku, Tatsuya Harada

NeRF

1

2

3



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng

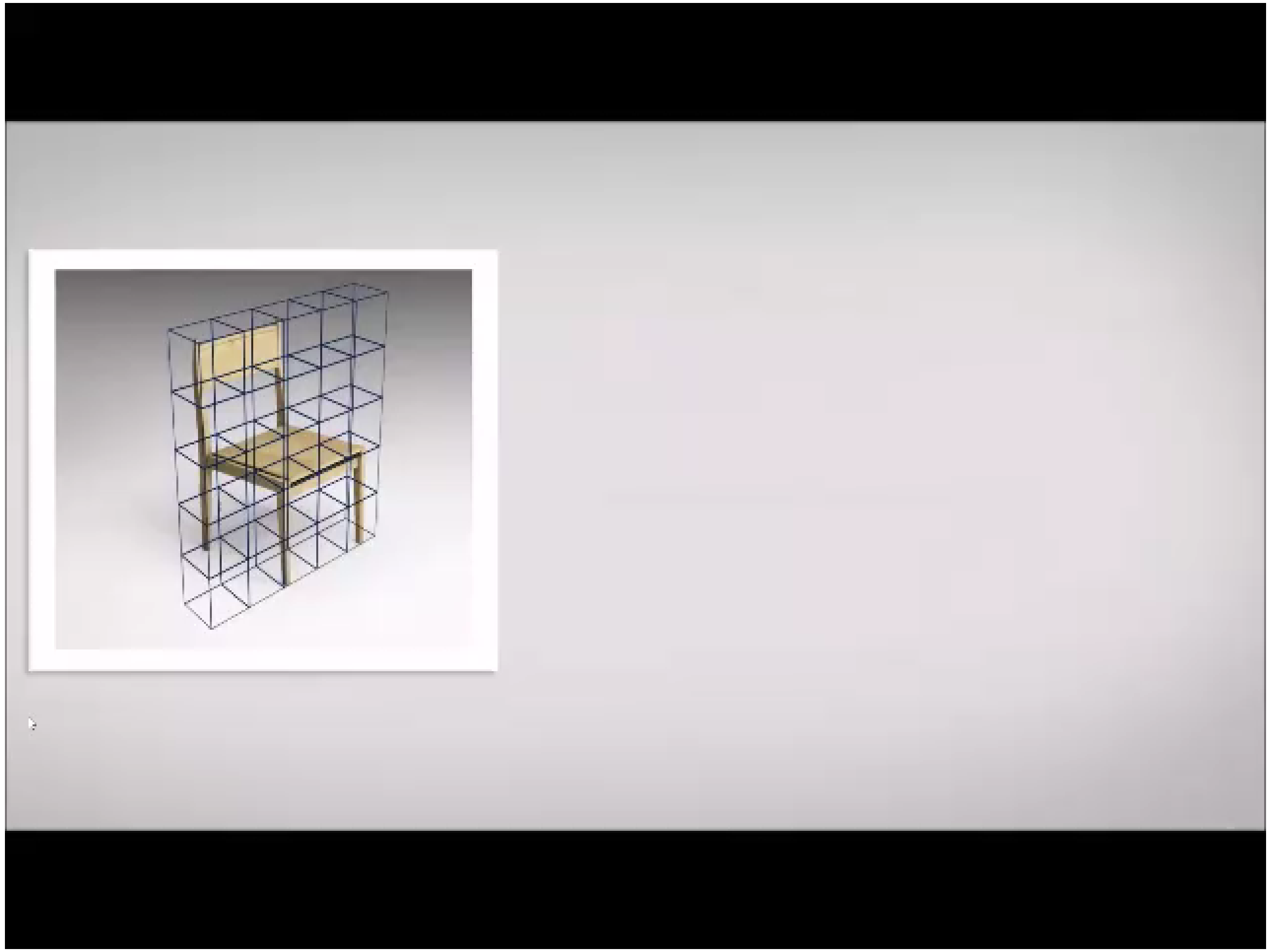
Neural Radiance Fields (NeRF) for unconstrained photo collection
Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, Daniel Duckworth

Mesh R-CNN

1

2

3



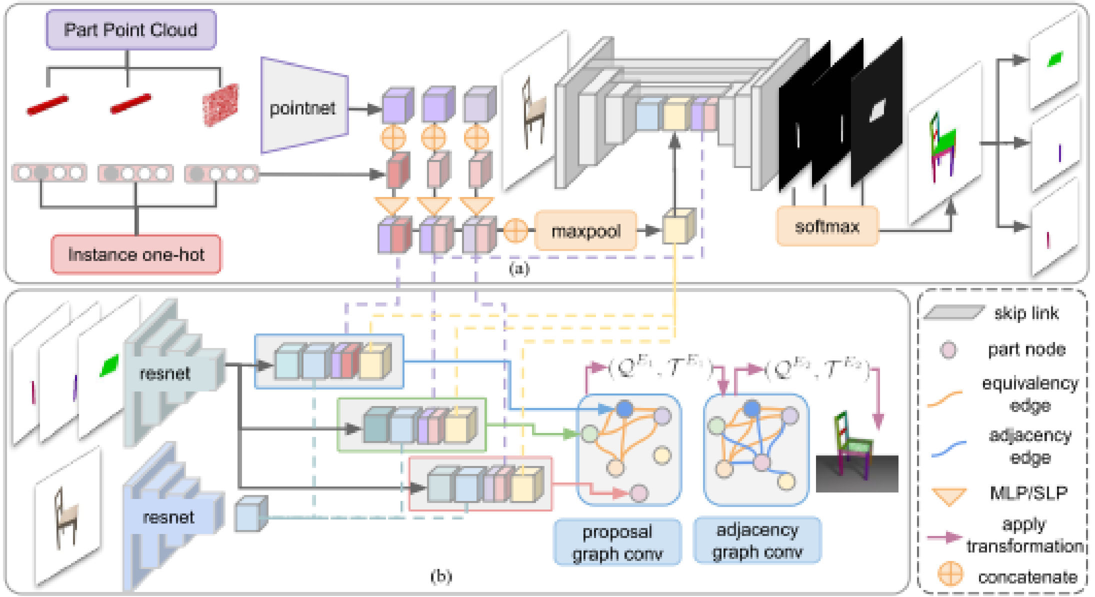
Mesh R-CNN Georgia Gkioxari ,Jitendra Malik, Justin Johnson

1

3D Part Assembly

2

3



Learning 3D Part Assembly from a Single Image Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas

Solution

3 Mesh R-CNN + 3D Part Assembly

Solution

1

2

3

4

Input Image

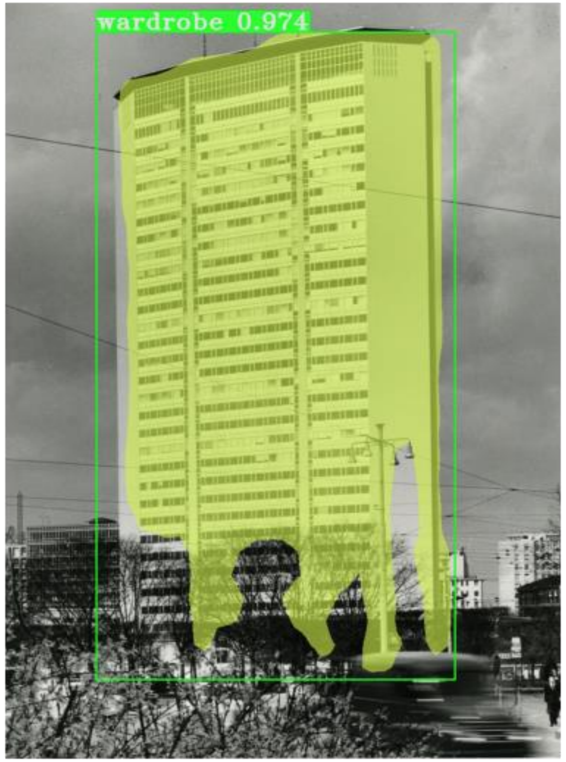
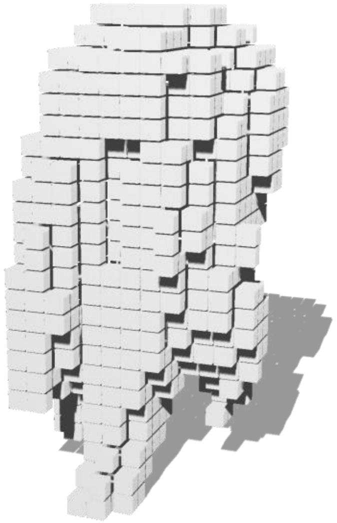


Image Recognition

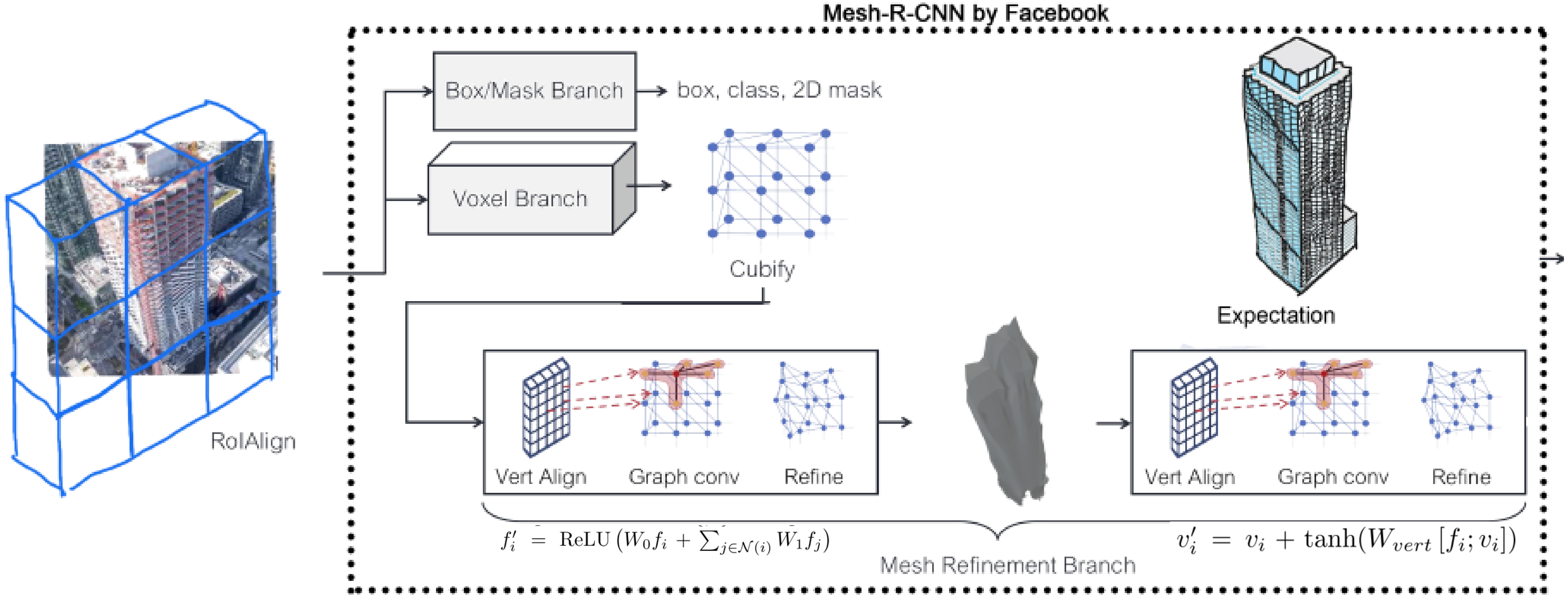


3D Voxels

3D meshes

Pytorch3D - Mesh R-CNN

$$\mathcal{L}_{\text{cham}}(P,Q) = |P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} \|p-q\|^2 + |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} \|q-p\|^2 \quad \mathcal{L}_{\text{norm}}(P,Q) = -|P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} |u_p \cdot u_q| - |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} |u_q \cdot u_p|. \quad \mathcal{L}_{\text{edge}}(V,E) = \frac{1}{|E|} \sum_{(v,v')\in E} \|v-v'\|^2,$$



Accelerating 3D Deep Learning with PyTorch3D Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, Georgia Gkioxari

Demo

https://cdinstitute.github.io/CompoNET/

```
# install dependencies: (use cu101 because colab has CUDA 10.1)
%env FORCE_CUDA=1
!pip install -U torch==1.5 torchvision==0.6 -f https://download.pytorch.org/whl/cu101/torch_stable.html
!pip install pyyaml==5.1 pycocotools>=2.0.1
!pip install -U fvcore
!pip install detectron2==0.1.3 -f https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.5/index.html
!pip install 'git+https://github.com/facebookresearch/pytorch3d.git'

import torch, torchvision
print(torch.__version__, torch.cuda.is_available(),torch.cuda.device_count())
!gcc --version
!python --version
!nvcc --version
!nvidia-smi
!python -m detectron2.utils.collect_env

!git clone https://github.com/facebookresearch/meshrcnn.git
!cd meshrcnn && pip install -e .

# pix3d data
!./meshrcnn/datasets/pix3d/download_pix3d.sh

from google.colab import files
uploaded = files.upload()

# Running Demo on random image of the pix3d dataset, change according to your need (chaneg 'test.jpg' to your file name)
# Be sure to upload the file on the correct folder

!python ./meshrcnn/demo/demo.py --config-file ./meshrcnn/configs/pix3d/meshrcnn_R50_FPN.yaml \
--input /content/test.jpg --output output_demo --onlyhighest MODEL.WEIGHTS meshrcnn://meshrcnn_R50.pth

filename = 'test.jpg' #check the name is the same as the file you uploaded
maskposition = '0_mask_sofa_1.000.png' # Make sure to change the file name with the appropriate mask change 'test/0_mask_sofa_1.000.png'

%matplotlib inline
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))

img = plt.imread("/content/" + filename)
plt.imshow(img)
plt.axis('off')
plt.show()

plt.figure(figsize=(10,10))
img = plt.imread("/content/output_demo/test/" + maskposition)
plt.imshow(img)
plt.axis('off')
plt.show()

name = 'sofa' # change file name accordingly

from google.colab import files
files.download('/content/output_demo/test/0_mesh_'+ name + '_1.000.obj')
```

CompoNET

CompoNET

View on GitHub

CompoNET: Geometric Deep Learning for Architecture

Open in Colab

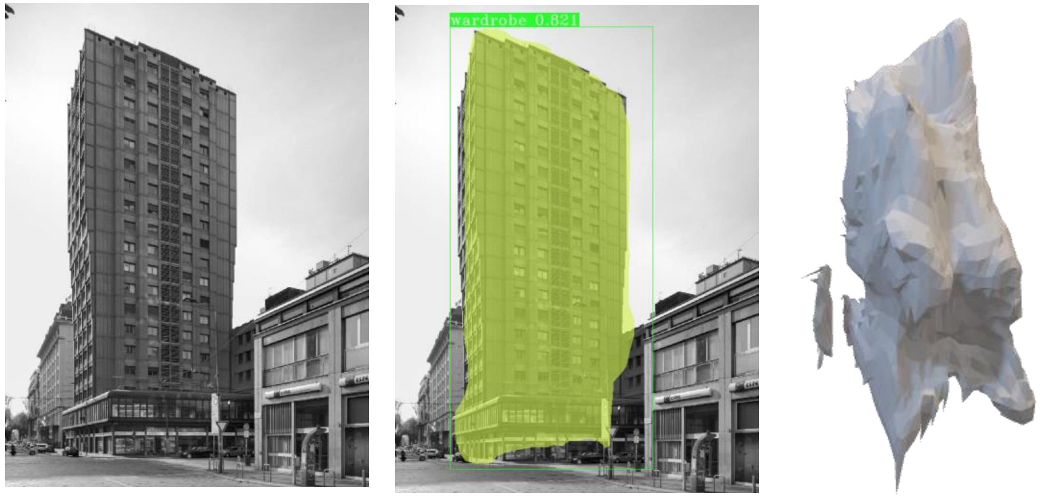
Authors

Alberto Tono, Cecilia Bolognesi, Amirhossein Ahmadnia, Meher Shashwat Nigam (2020)
alberto.tono@cd.institute
cecilia.bolognesi@polimi.it
amirhossein.ahmadnia@mail.polimi.it
meher.shashwat@students.iiit.ac.in

Examples

1

2 Torri Turati Muzio



3

Torre al Parco Magistretti

4 Torre Breda Mattioni

5

Via Quadronno Mangiarotti

Results

- Limitations
  - Current segmentation not optimize for buildings
  - Lack of an heterogenous dataset
  - Poor evaluation metrics
  - Lack of good generalization
  - Lack of main topological properties

1 # Future Research

2 - Optimize Loss Function, Evaluation Metrics and Mesh Refinement Branch

3 - Creation of a comprehensive and heterogeneous Dataset

4 - Introducing BIM

5 - Establishment of an Open Collaboration Framework

Thanks

1

2


Torri Turati Muzio


Torre Breda Mattioni


Torre al Parco Magistretti

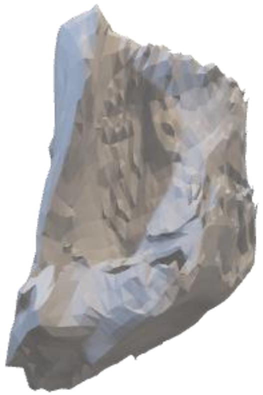

Ponti- Pirellone


Palazzo Montecatini Ponti


Via Quadronno Mangiarotti

3

4



5