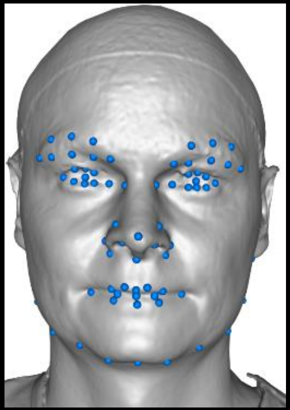
Geometric Deep Learning on 3D Meshes an overview

Rasmus R. Paulsen
DTU Compute

Based on contributions from (among others):

Kristine Aavild Juhl
Christian Keilstrup Ingwersen
Patrick Møller Jensen
Mathias Micheelsen Lowes
Bjørn Marius Schreblowski Hansen
Anders Bjorholm Dahl
Vedrana Andersen Dahl



Who is this aimed at?

- The ideal audience
  - Limited practical experience with geometrical deep learning
  - Has a good understanding of basic convolutional neural networks
    - Has seen the U-net before
  - Might come in a situation where your data is actually 3D meshes or have been *magicked* into 3D meshes
  - Would like to do surface based classification or labelling / segmentation
  - Lacks a good starting point
    - Which approach is good for my data

When poll is active, respond at **pollev.com/rasmuspaulse538**

What is your experience with geometric deep learning?

This is the first time I hear about it

I have superficial knowledge about the field

I have read several articels about it

I have tested an existing framework

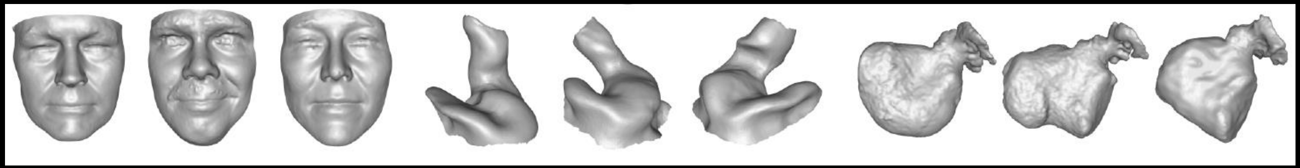
I have adapted an existing framework to my own data

I have coded my own framework

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

What's in it for me?

- You will (hopefully) get an overview of different approaches to work with 3D meshes
- Some understanding of the strengths and weaknesses of the different methods
  - How invariant the methods are to geometric transformation (translation, rotations etc)
  - How large meshes can they process?
  - What are the restriction with regards to geometry/topology
  - How do they handle noise?



When poll is active, respond at **pollev.com/rasmuspaulse538**

What is my interest in the field?

I am here for the ECTS, the social network and of general interest

I am working with data that might benefit from geometric deep learning

I have a theoretical interest in the field and would like to advance the theory in the field

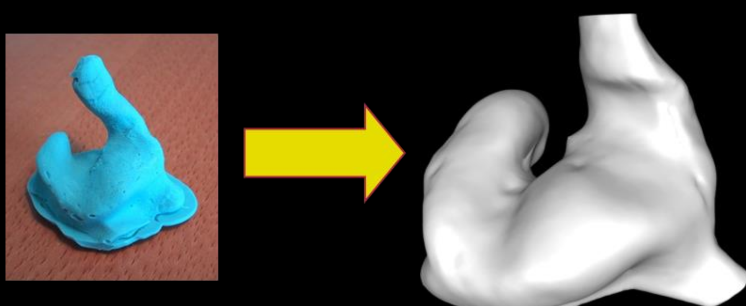
Something else

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Surfaces – where do they come from?

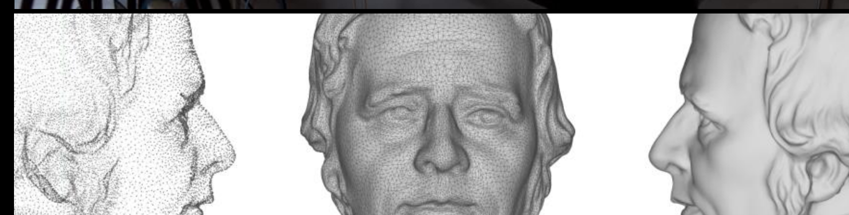
Direct surface scanning using a Canfield Vectra facial scanner.

Object scanners



An ear impression scanned by a 3Shape scanner.

Probably one of the most scanned anatomies in the world

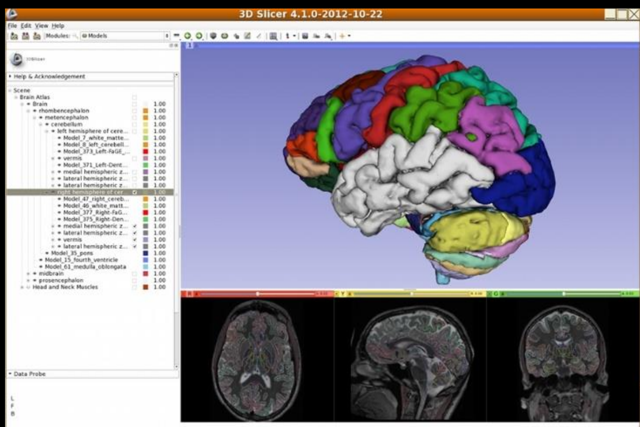


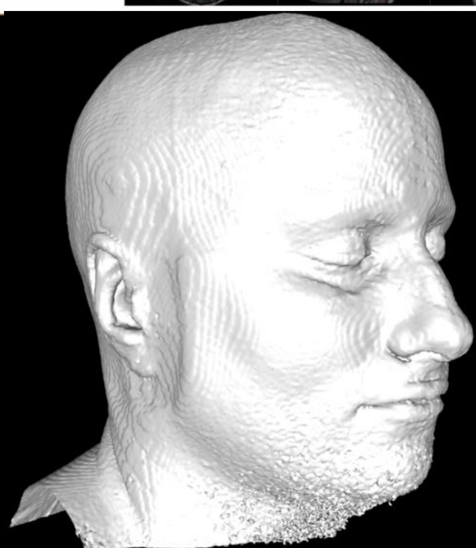
The founder of DTU – H. C. Ørsted

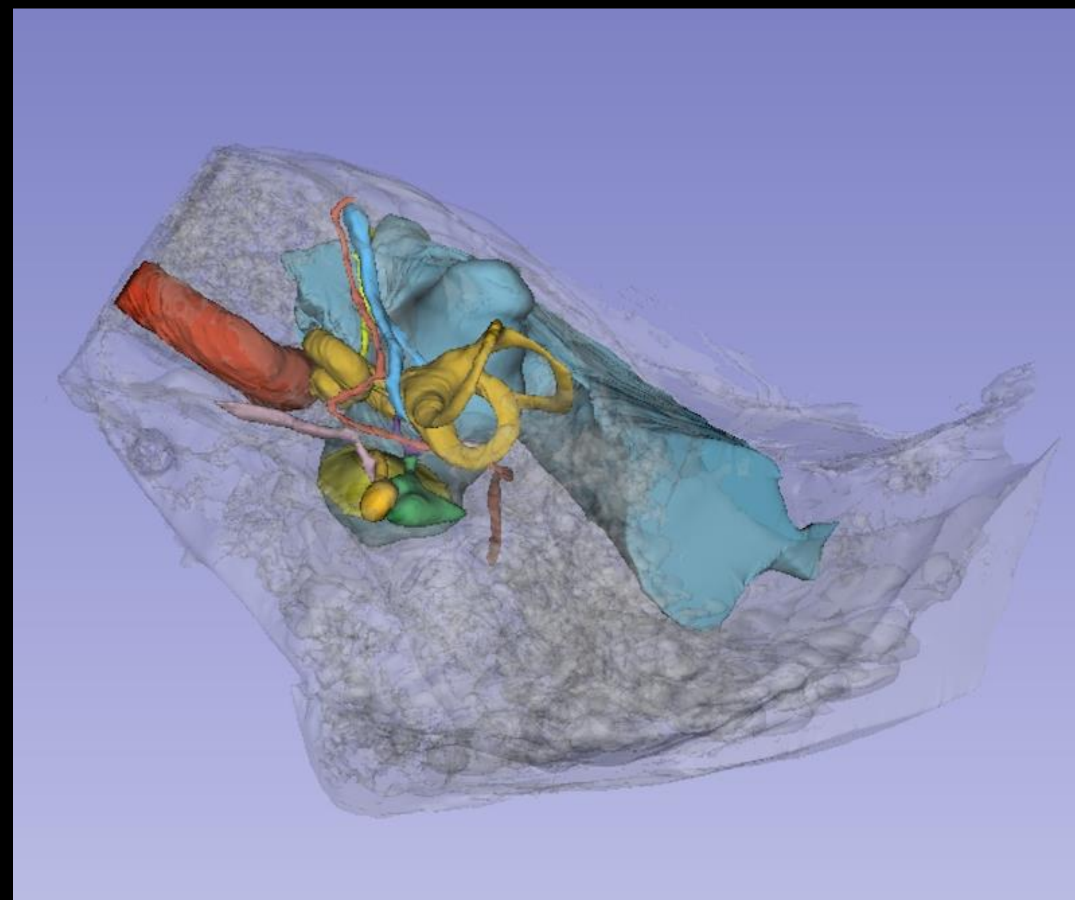
Scanned by Dolores Messer with a custom built structured light scanner at DTU Compute

Eiríksson et al. "Precision and accuracy parameters in structured light 3-D scanning." International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 5 (2016)

Iso-surfaces or pixel-wise classifications





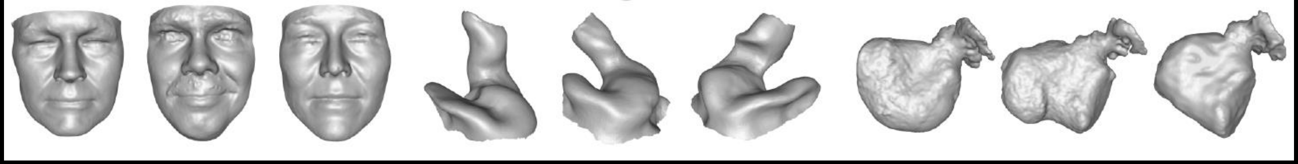


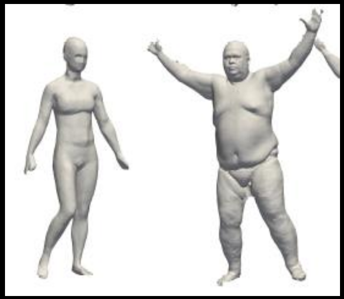
CAD Models

Chang, Angel X., et al. "Shapenet: An information-rich 3d model repository.” (2015).

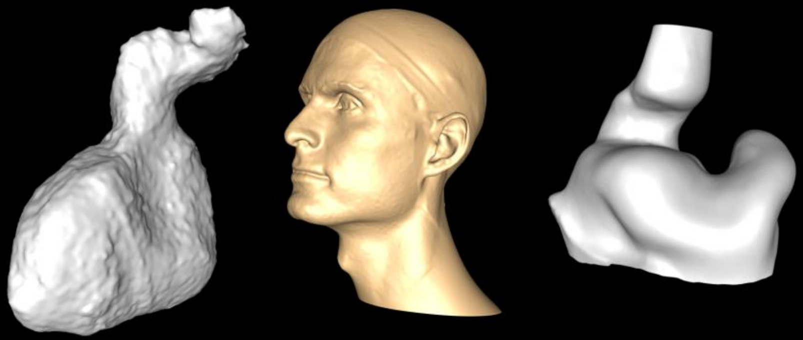
Important properties of meshes

- Rotational aspects (geometric invariances)
- Size (number of vertices and faces)
- Topology and if it is ”manifold”
- Mesh sampling and noise properties





Translation and rotational aspects

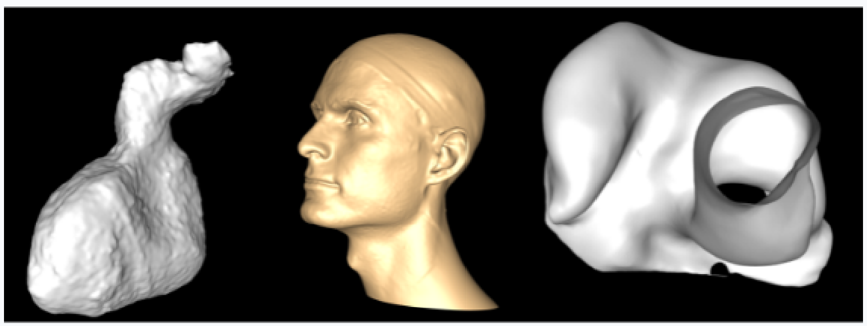


- Does it make sense to have a "canonical orientation" of your objects?
- Does the method require that the objects are pre-oriented?
- Translation is often fixed by aligning center-of-mass
  - Not a universal solution

Mesh sizes

Type	Vertices	Faces
Shapenet model (CAD)	Hundreds (guess)	Hundreds (guess)
Facial scan with accuracy~0.5 mm	110.000	35.440
Left atrium from CT scan (voxel size 0.50mm^3) (iso-surface)	35.000	65.000
Scanned H. C. Ørsted (accuracy 150 mikrometer)	1.375.930	2.751.840
Full head model with accuracy ~1 mm	450.000	830.000
FAUST human body (processed)	6.890	

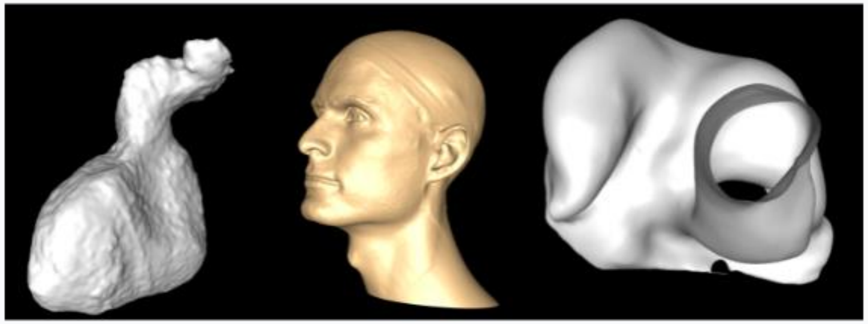
What are the topogogical equivalences of the three meshes?



- Sphere, Sphere, Plane
- Sphere, Plane, Plane
- Sphere, Plane, Tube
- Plane, Sphere, Tube
- Sphere, Sphere, Sphere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

What are the topogogical equivalences of the three meshes?



Sphere, Sphere, Plane

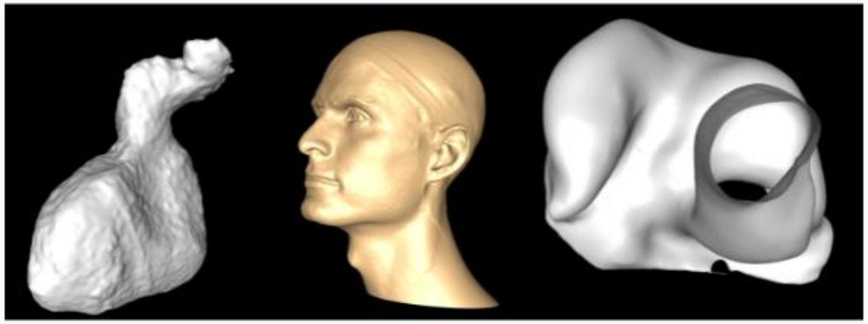
Sphere, Plane, Plane

Sphere, Plane, Tube

Plane, Sphere, Tube

Sphere, Sphere, Sphere

What are the topogogical equivalences of the three meshes?



Sphere, Sphere, Plane

Sphere, Plane, Plane

Sphere, Plane, Tube

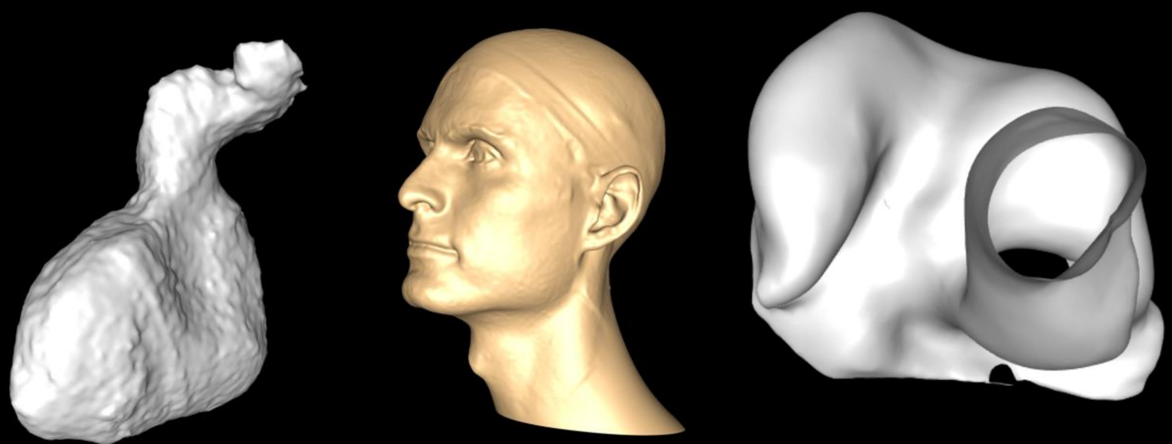
Plane, Sphere, Tube

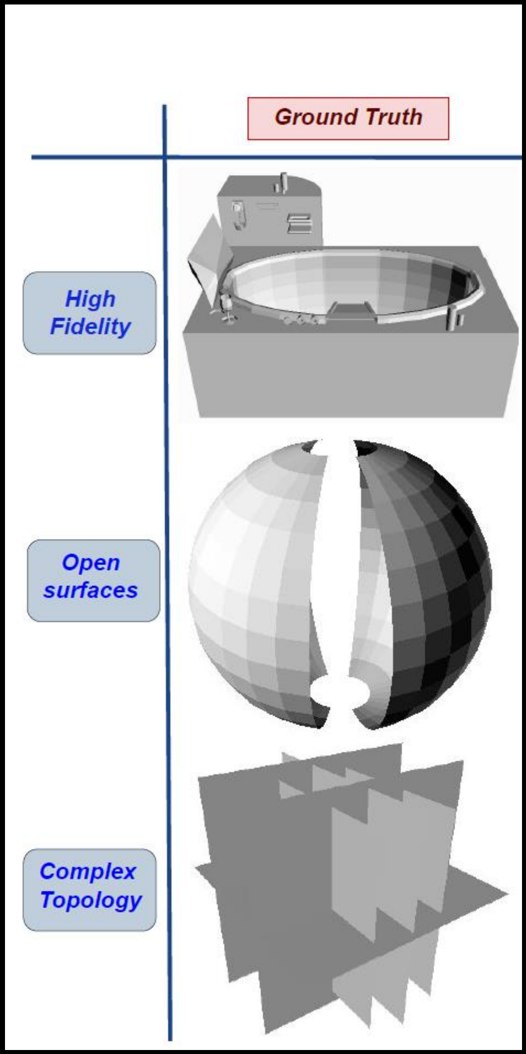
Sphere, Sphere, Sphere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Mesh topology

- Topologically equivalent to a
  - sphere, plane, tube, donut?
  - or something far far beyond?
- Is it ”manifold” ?

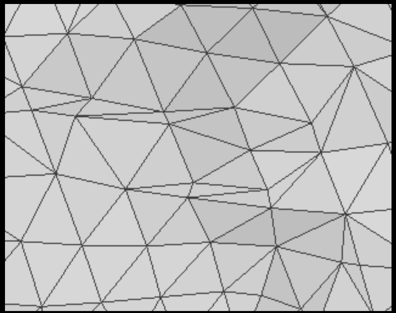


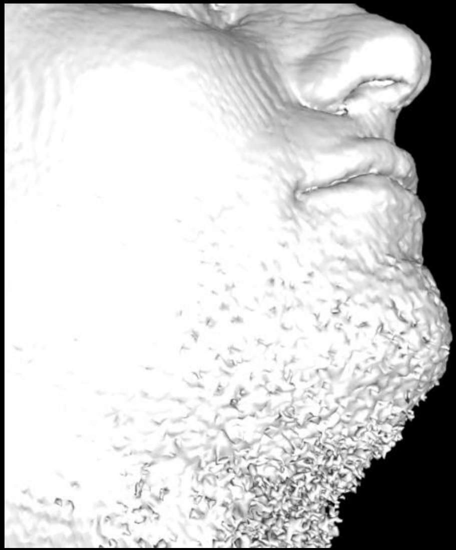


Venkatesh, Rahul, et al. "DUDE: Deep Unsigned Distance Embeddings for Hi-Fidelity Representation of Complex 3D Surfaces." *arXiv:2011.02570* (2020).

Mesh sampling and noise?

- Are the vertices sampled equally over the underlying surface?
- Are the faces/triangles well shaped?
  - Classical *marching cubes* makes notoriously bad aspect ratio triangles
- What is the nature of the sampling noise?
  - Outliers, Gaussian or something else?



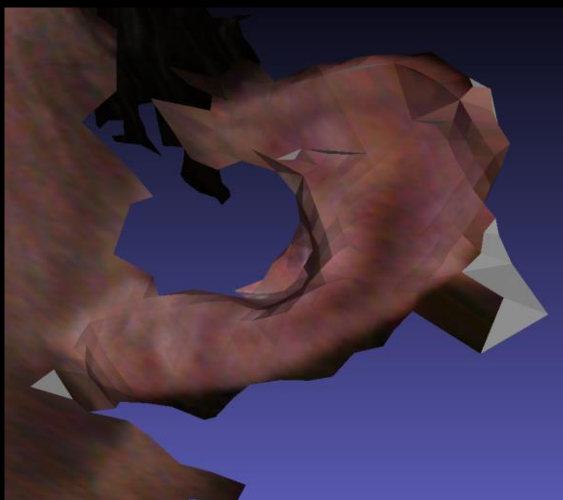


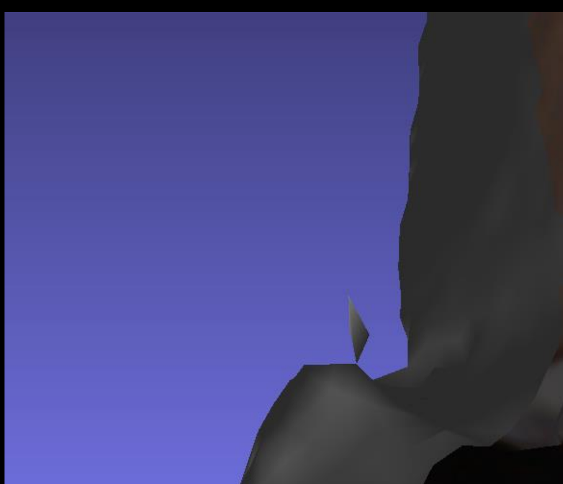
A mesh biopsy

- Raw facial scan from BU-3DFE – a reference dataset
- “Mesh in the wild”
  - representative for current facial scanners
- 106.320 vertices and 35.440 faces

”A 3D Facial Expression Database For Facial Behavior Research”
by Lijun Yin; Xiaozhou Wei; Yi Sun; Jun Wang; Matthew J. Rosato, 7th International Conference on Automatic Face and Gesture Recognition, 10-12 April 2006 P:211 - 216

A mesh biopsy





- Looks topologically to be a plane
  - but it is not
- Flipped triangles
- Non-manifold parts
- Complex noise issues
- A face has a canonical orientation
  - But facial scanners have many different coordinate systems

Do topology and artefacts matter?

- Quite a lot actually
  - A lot of the current methods have severe restrictions on topology and if the surfaces are manifold

- A crude comparison
  - Imagine your 2D CNN would crash and burn because of one single bad pixel due to a dead CCD cell

- A typical solution – preprocess the mesh so it is nice and clean
  - Often needs a specific solution for each dataset
  - Large risk of removing / smoothing out important information

My experience with the U-net

Never heard of it

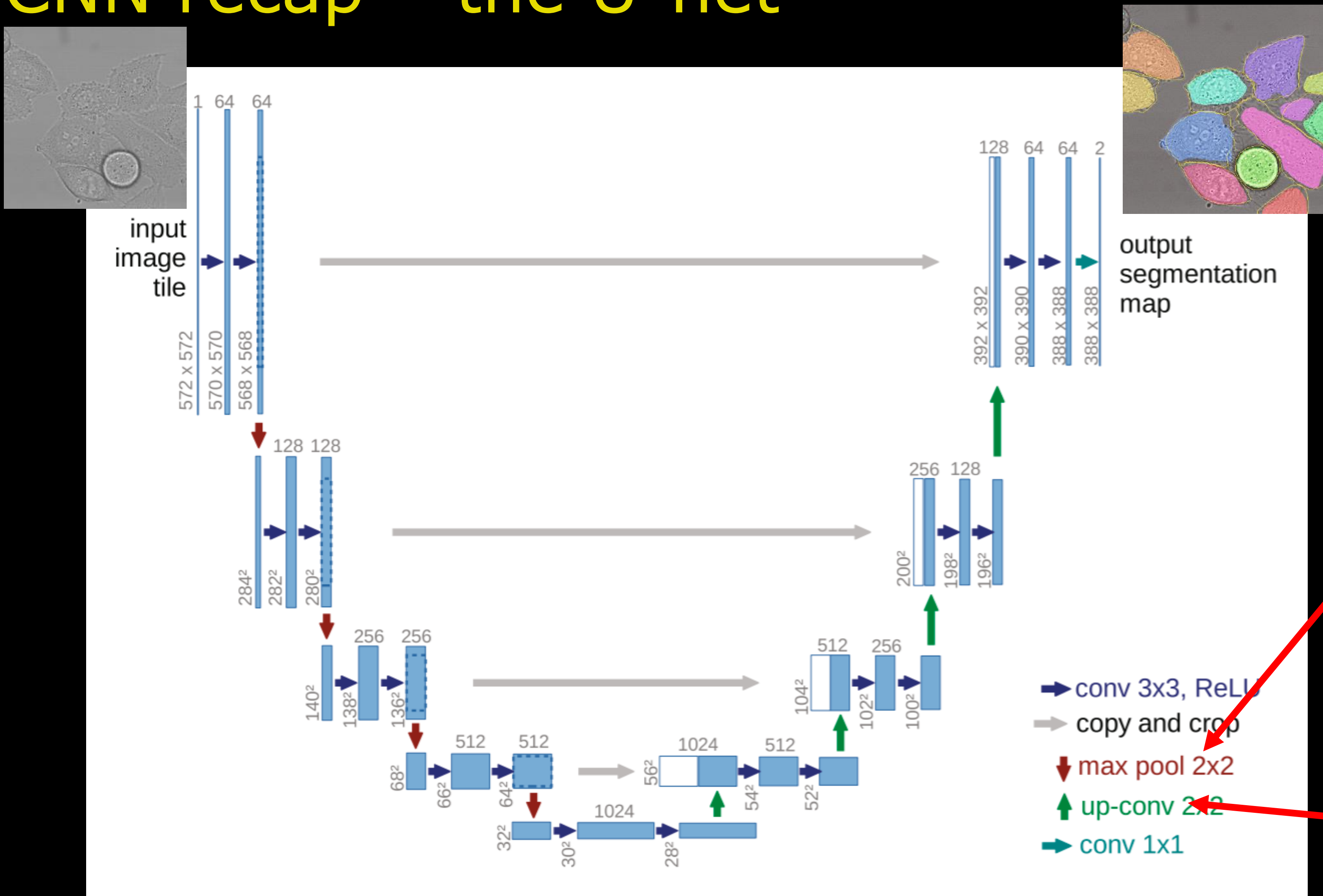
I have superficial knowledge of the U-net

I have read several papers where the U-net is used

I have tried a pre-made U-net

I have coded my own custom version of the U-net

CNN recap – the U-net

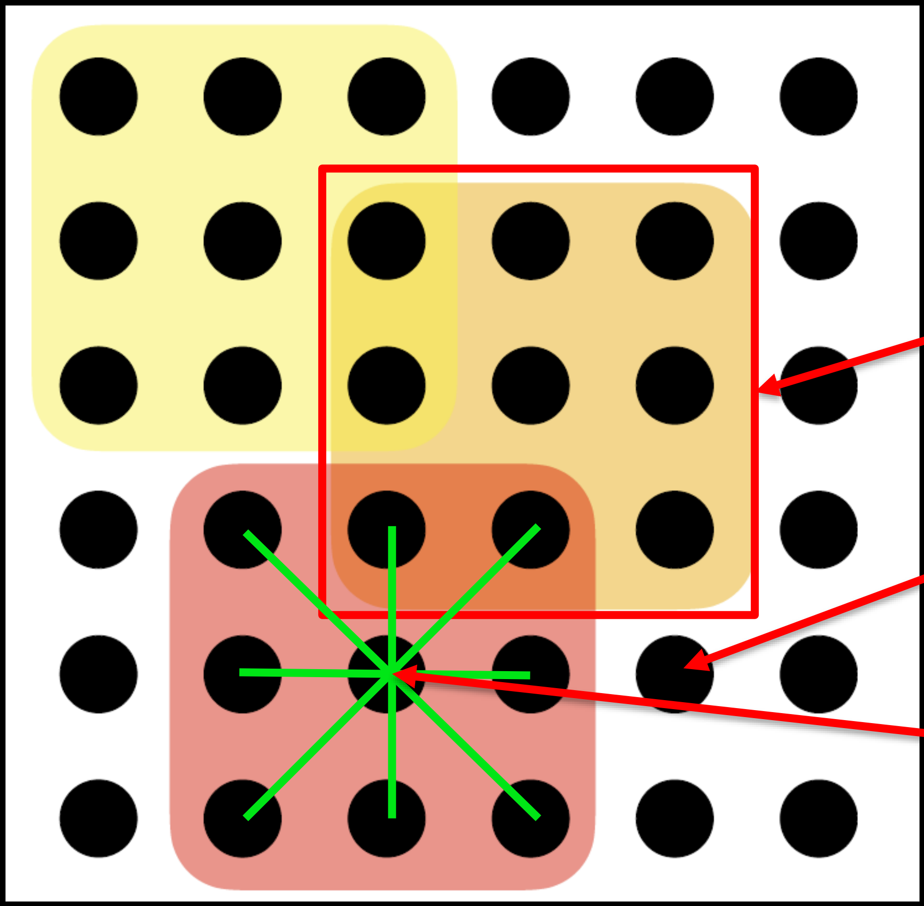


Pooling

Convolution

Ronneberger et al. "U-net: Convolutional networks for biomedical image segmentation." *MICCAI.* 2015.

Convolution – a conceptual heads-up



Your data – an image, a mesh, a graph or something more exotic

A kernel – containing (learnable) weights

For each ”node” in your data you have values

For each ”node” in your data you have a neighborhood that should be ”covered” by the kernel

But first something completely different!

Approaches covered in the following

- Multi-view rendering approaches
- Volumetric approaches
- Methods that define convolutions on meshes
- Methods based on implicit representations of meshes.
  - For example implicit functions on grids and signed/un-signed distance fields
- Hybrid methods based on mesh operations for convolutions and pooling

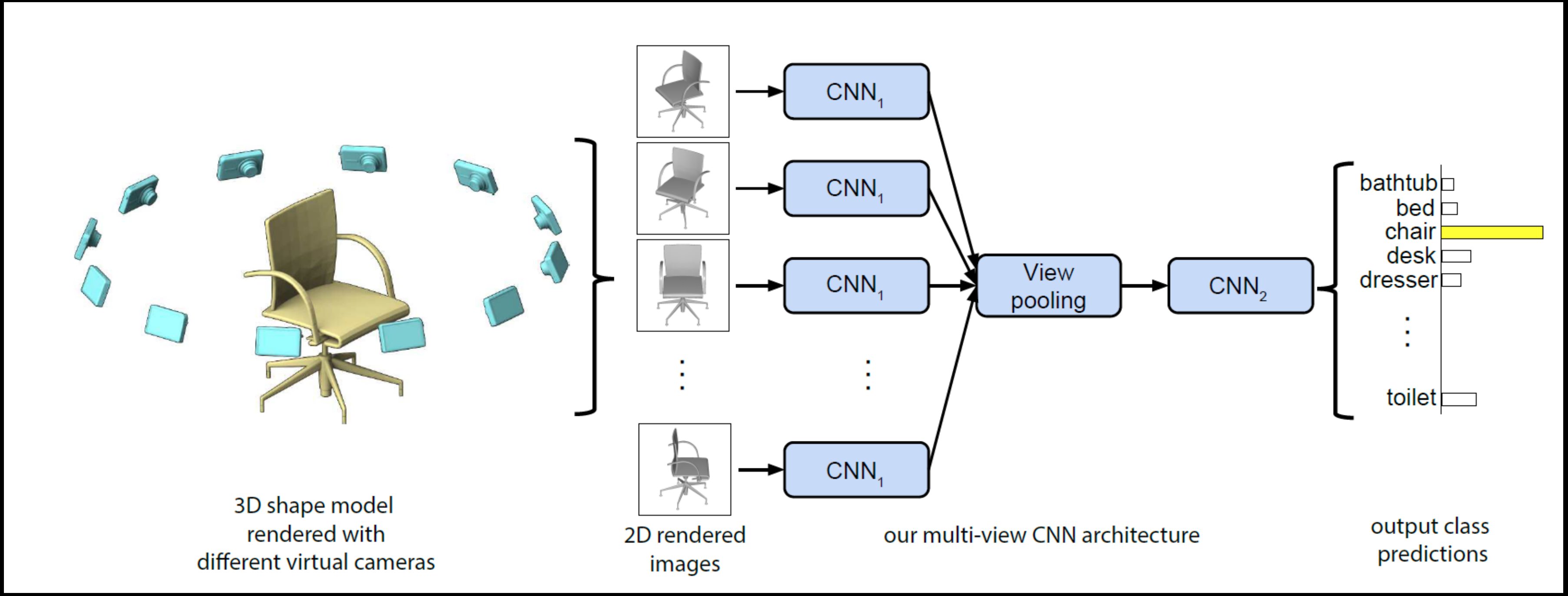
Disclaimer: It will mostly be a conceptual overview

I am certainly not a specialist on all approaches.

Multi-view Convolutional Neural Networks for 3D Shape Recognition

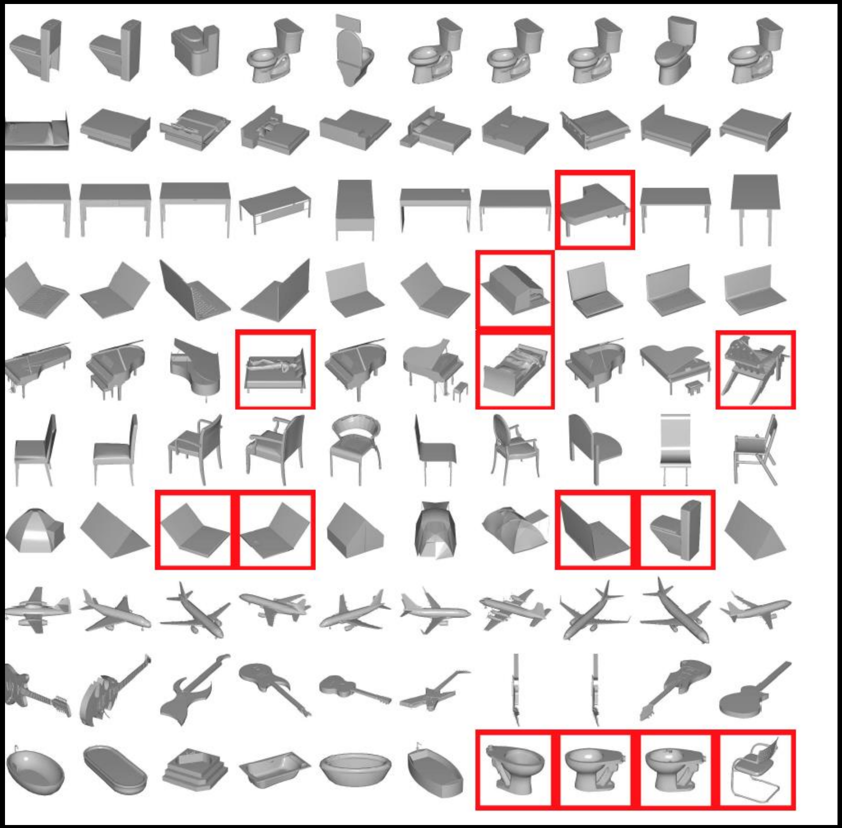
Hang Su Subhransu Maji Evangelos Kalogerakis Erik Learned-Miller
University of Massachusetts, Amherst
{hsu,smaji,kalo,elm}@cs.umass.edu

1900 google scholar citations per August 2021



Su, Hang, et al. "Multi-view convolutional neural networks for 3d shape recognition." *Proceedings of the IEEE international conference on computer vision*. 2015.

Multi-view convolutional neural networks for 3d shape recognition

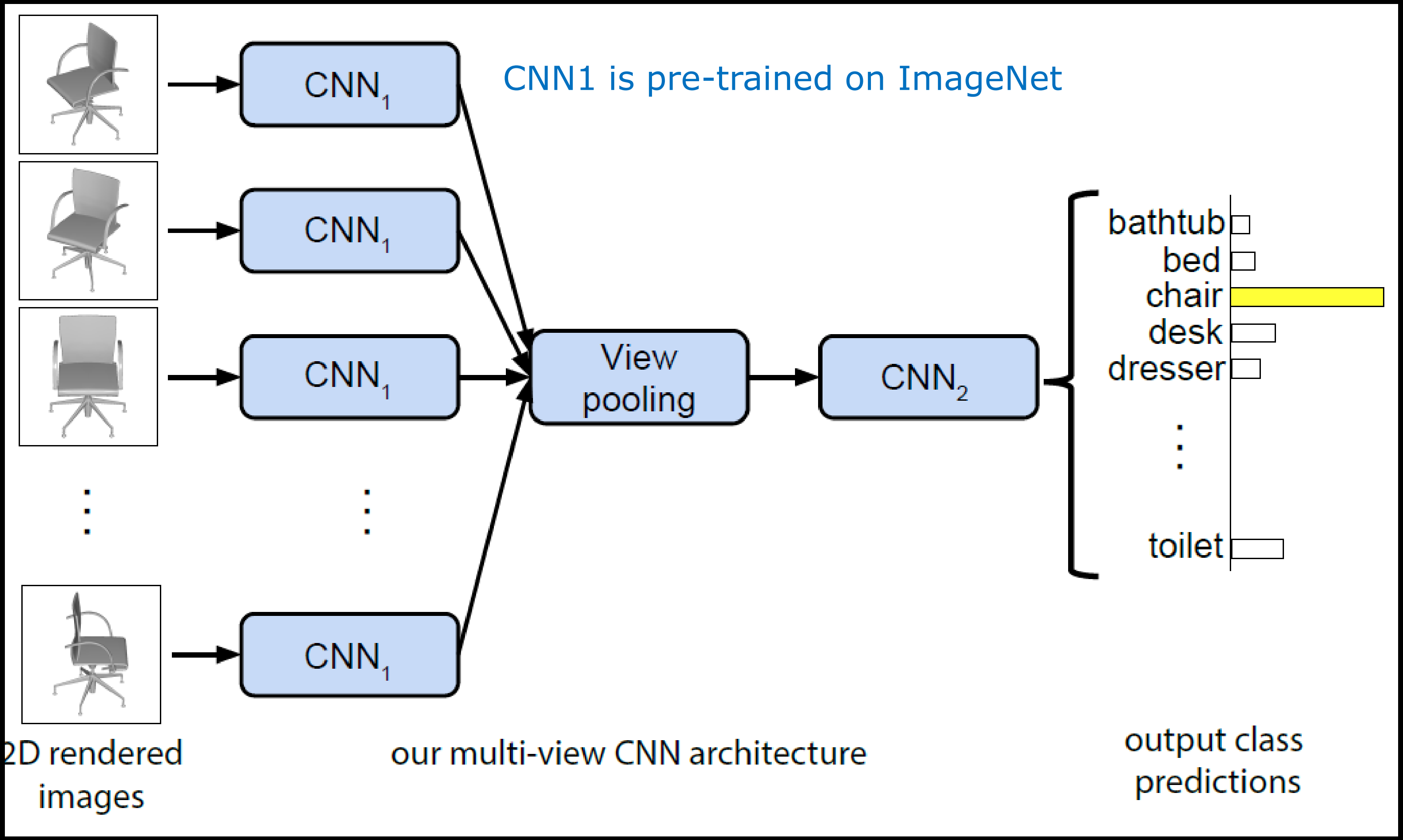


- Object classification based on 3D shapes
- Rendering pipeline
- Standard 2D CNN to do the classification

Multi-view convolutional neural networks for 3d shape recognition – rendering setup

- 12 positions with rotations around the z-axis
- 80 views
  - 20 vertices of an icosahedron enclosing the shape
  - 4 rotations around camera axes

Multi-view convolutional neural networks for 3d shape recognition – network



Multi-view convolutional neural networks for 3d shape recognition – results

Princeton ModelNet

- 128K 3D CAD models
- 662 categories

Modelnet40

- 12K models
- 40 categories

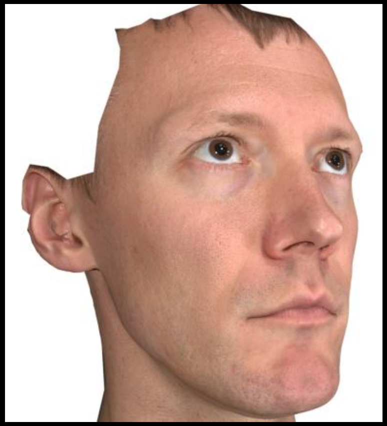
Method	Training Config.			Test Config.	Classification	Retrieval
	Pre-train	Fine-tune	#Views	#Views	(Accuracy)	(mAP)
(1) SPH [16]	-	-	-	-	68.2%	33.3%
(2) LFD [5]	-	-	-	-	75.5%	40.9%
(3) 3D ShapeNets [37]	ModelNet40	ModelNet40	-	-	77.3%	49.2%
(4) FV	-	ModelNet40	12	1	78.8%	37.5%
(5) FV, 12×	-	ModelNet40	12	12	84.8%	43.9%
(6) CNN	ImageNet1K	-	-	1	83.0%	44.1%
(7) CNN, f.t.	ImageNet1K	ModelNet40	12	1	85.1%	61.7%
(8) CNN, 12×	ImageNet1K	-	-	12	87.5%	49.6%
(9) CNN, f.t.,12×	ImageNet1K	ModelNet40	12	12	88.6%	62.8%
(10) MVCNN, 12×	ImageNet1K	-	-	12	88.1%	49.4%
(11) MVCNN, f.t., 12×	ImageNet1K	ModelNet40	12	12	89.9%	70.1%
(12) MVCNN, f.t.+metric, 12×	ImageNet1K	ModelNet40	12	12	89.5%	**80.2%**
(13) MVCNN, 80×	ImageNet1K	-	80	80	84.5%	36.8%
(14) MVCNN, f.t., 80×	ImageNet1K	ModelNet40	80	80	**90.1%**	70.4%
(15) MVCNN, f.t.+metric, 80×	ImageNet1K	ModelNet40	80	80	**90.1%**	79.5%

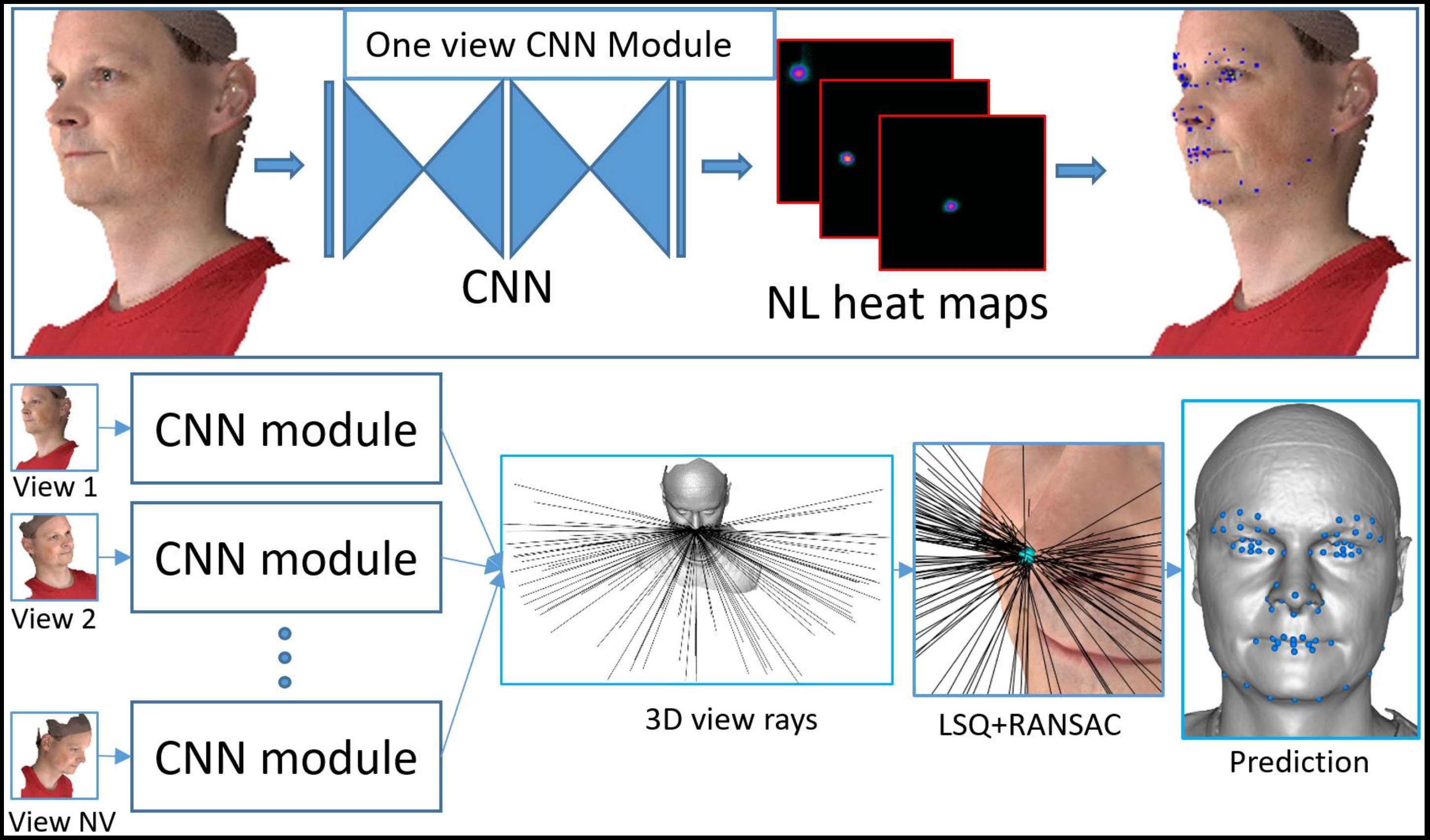
* f.t.=fine-tuning, metric=low-rank Mahalanobis metric learning

Multi-view convolutional neural networks for 3d shape recognition – some observations

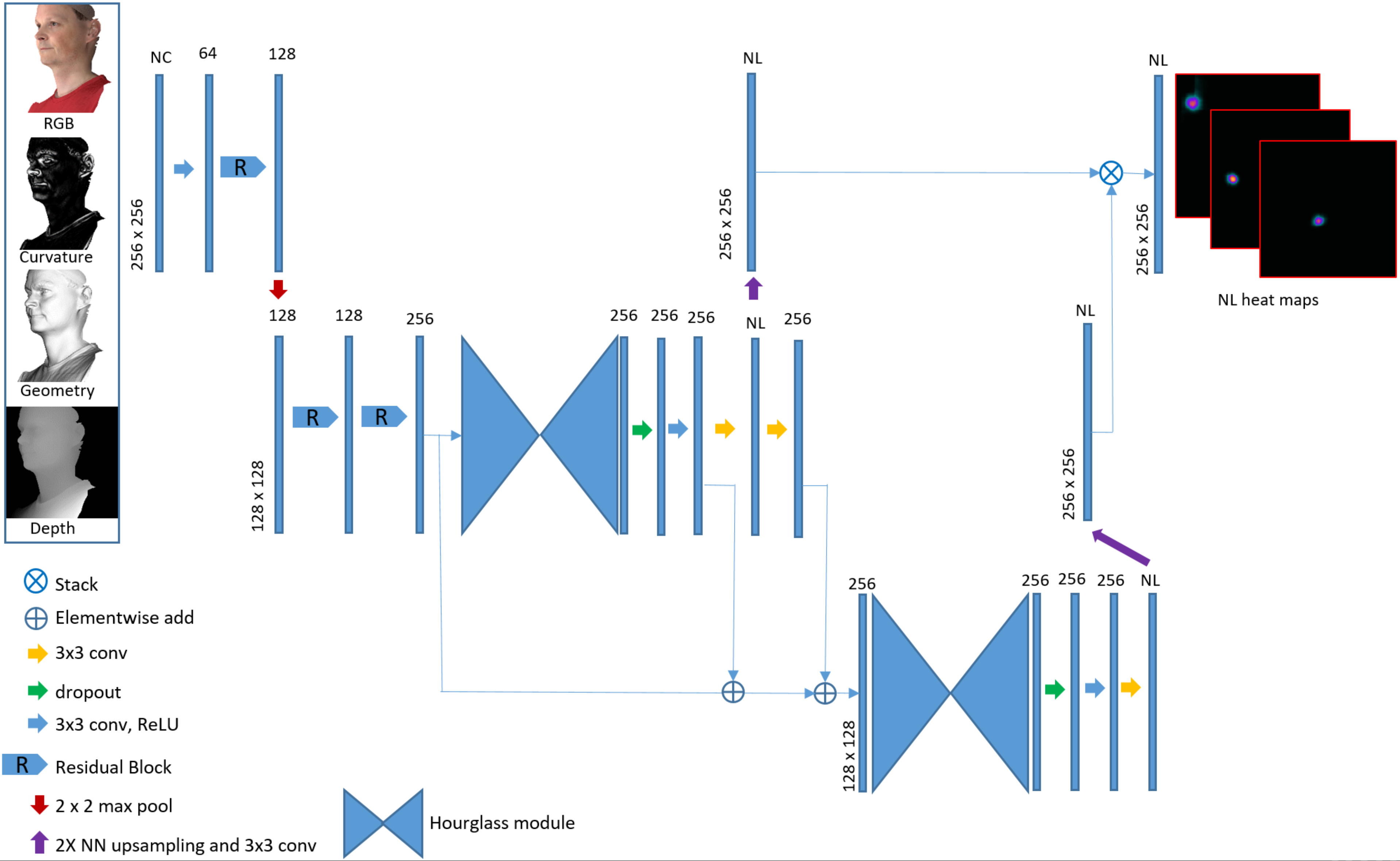
- If you can render your object – you can classify it
  - Robust to topology variations, large mesh sizes, noise
- Pre-aligning an object to a canonical orientation is ill-posed
  - the view sequence is somewhat arbitrary
  - Only partially rotationally invariant

Multi-view CNN for landmark prediction

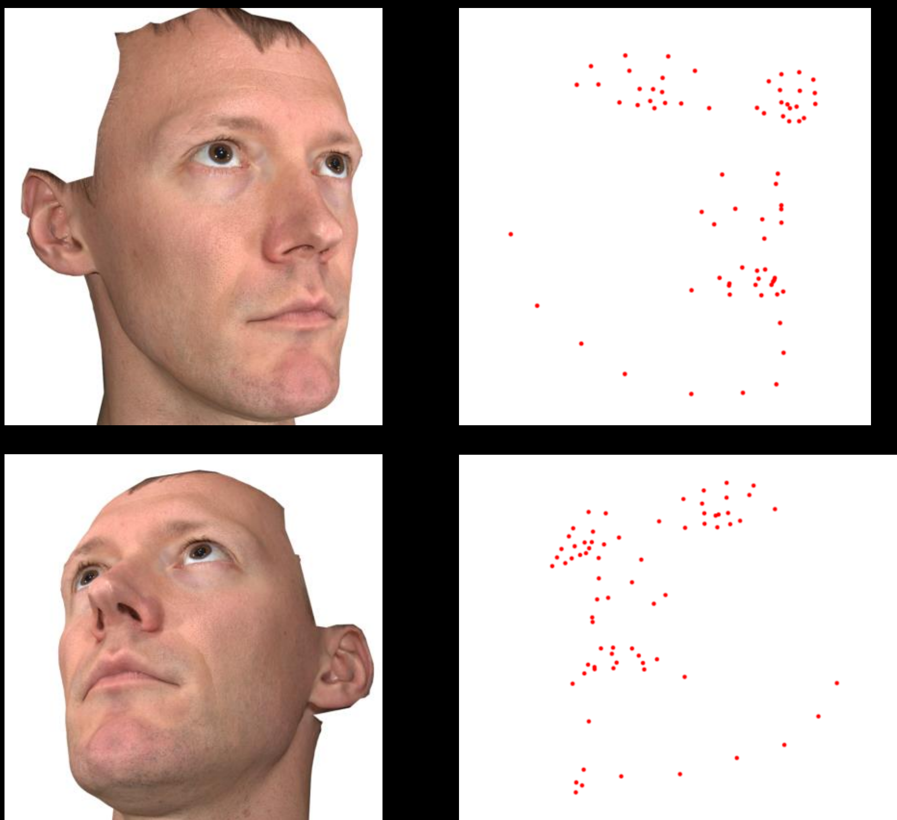




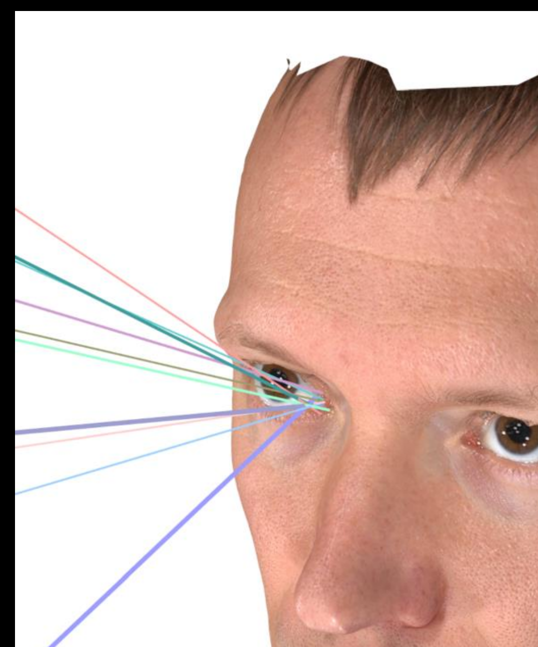
Paulsen et al. "Multi-view consensus CNN for 3D facial landmark placement.“. Proc. Asian Conference on Computer Vision. (2018)


RGB
Curvature
Geometry
Depth
NC
64
128
R
256 x 256
128
128
256
R
R
128 x 128
256
256
256
NL
256
NL
256 x 256
NL
256 x 256
NL heat maps
NL
256 x 256
256
256
256
256
NL
128 x 128
Stack
Elementwise add
3x3 conv
dropout
3x3 conv, ReLU
R
Residual Block
2 x 2 max pool
2X NN upsampling and 3x3 conv
Hourglass module

3D landmark prediction



- Given a set of rendered faces
- 2D landmark positions are estimated
- A predicted landmark in 2D corresponds to a line in space



What can RANSAC do for me here?

Sample random positions in space for view directions

Render coherent images of skin

Robustly estimate a line crossing avoiding outlier influence

Effectively computing intersection between rays and a triangulated surface

What can RANSAC do for me here?

Sample random positions in space for view directions

Render coherent images of skin

Robustly estimate a line crossing avoiding outlier influence

Effectively computing intersection between rays and a triangulated surface

When poll is active, respond at **pollev.com/rasmuspaulse538**

What can RANSAC do for me here?

Sample random positions in space for view directions

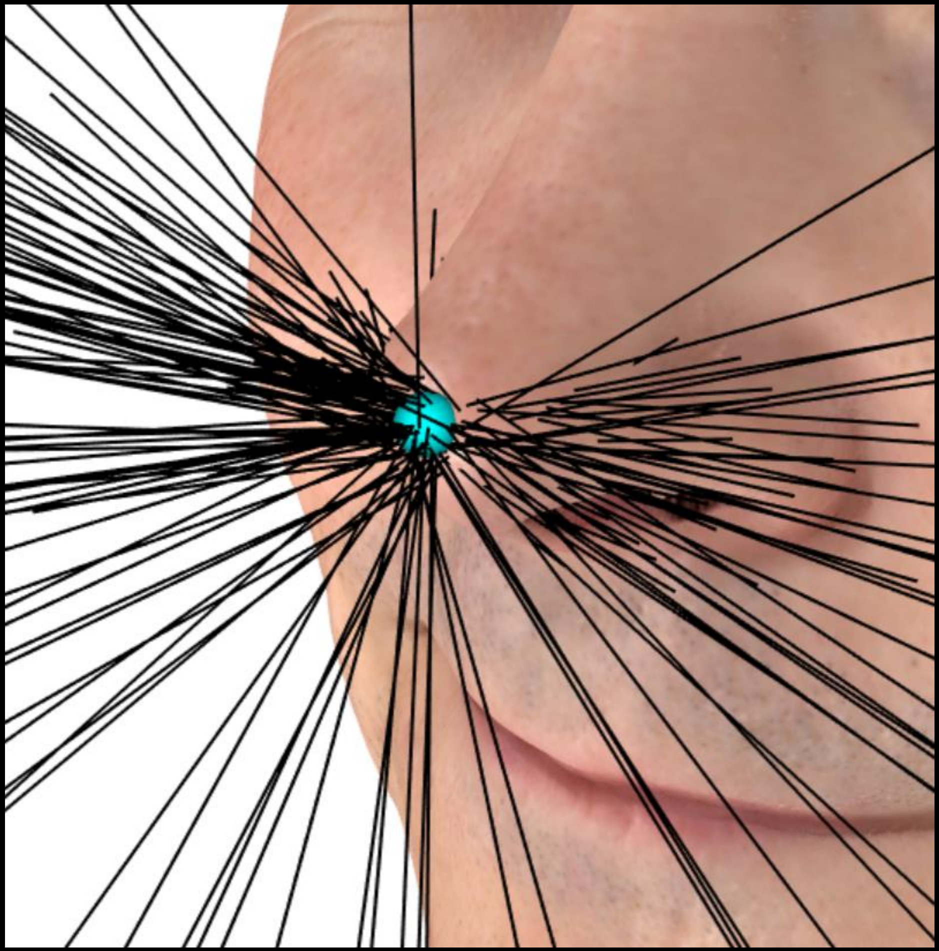
Render coherent images of skin

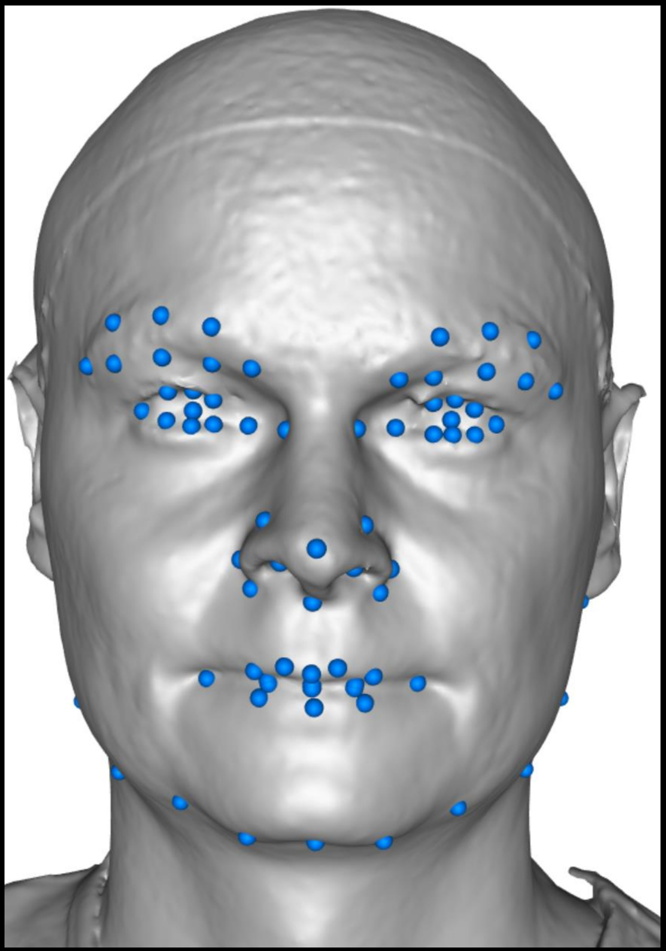
Robustly estimate a line crossing avoiding outlier influence

Effectively computing intersection between rays and a triangulated surface

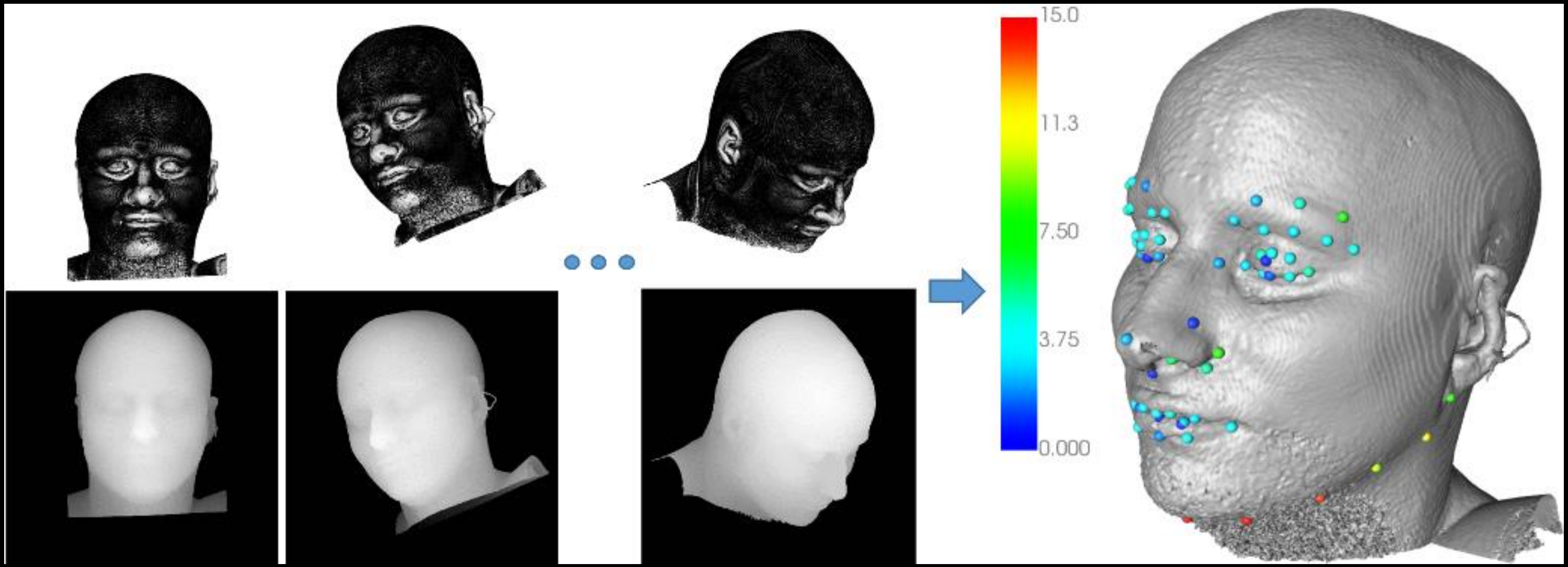
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Least squares and RANSAC





Using trained network on MR data

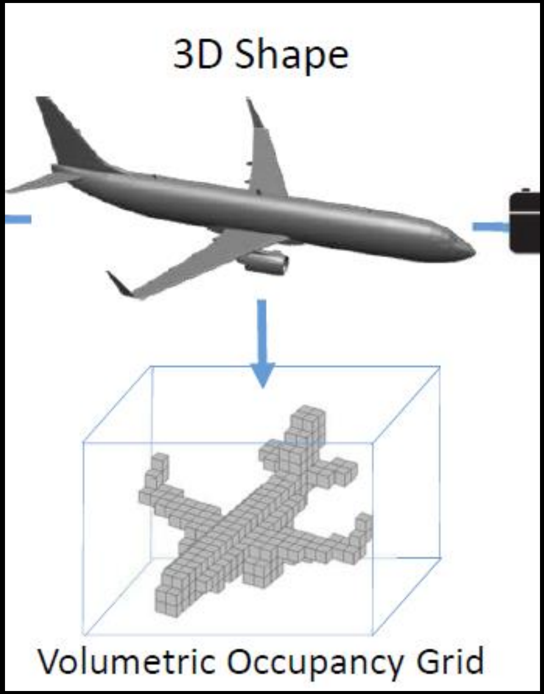


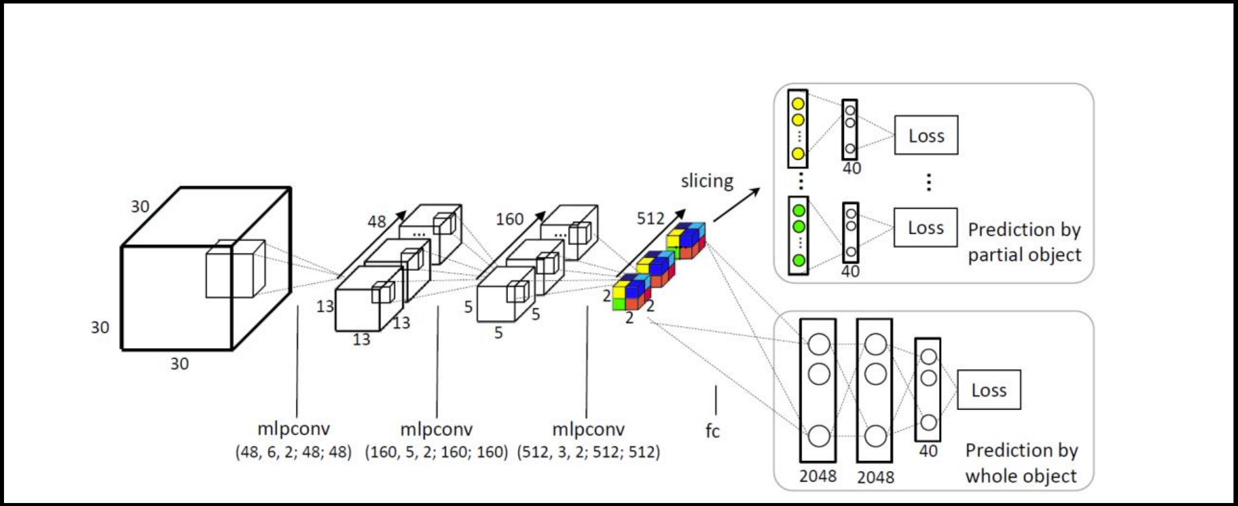
Trained on the z-buffer / distance map

Works with significant amount of surface noise

http://shapeml.compute.dtu.dk/

Volumetric CNN for object classification - occupancy representation



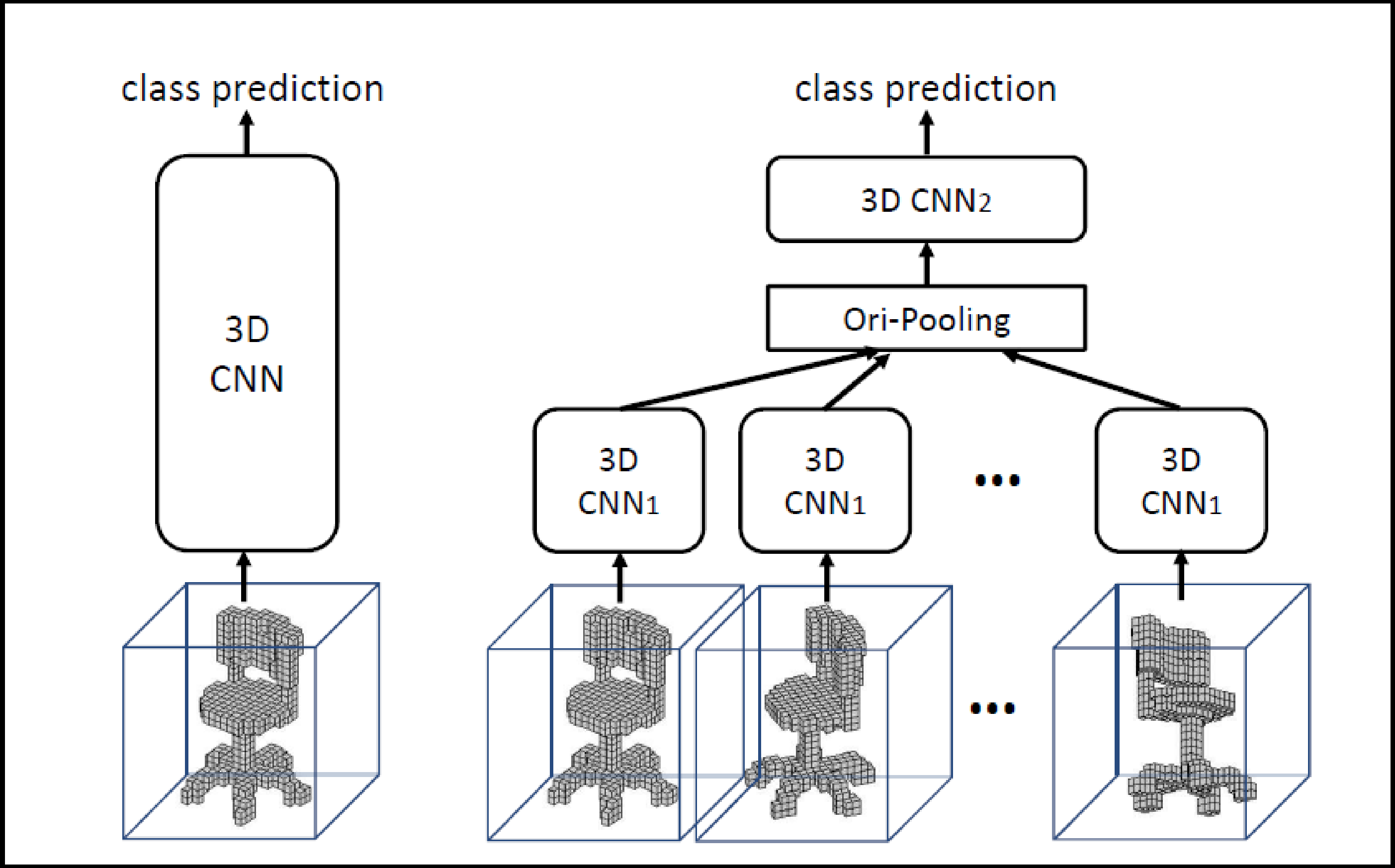


30 x 30 x 30 occupancy grid

Qi, Charles R., et al. "Volumetric and multi-view cnns for object classification on 3d data." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Wu, Zhirong, et al. "3d shapenets: A deep representation for volumetric shapes." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Volumetric CNN for object classification



Volumetric CNN for object classification – some observations

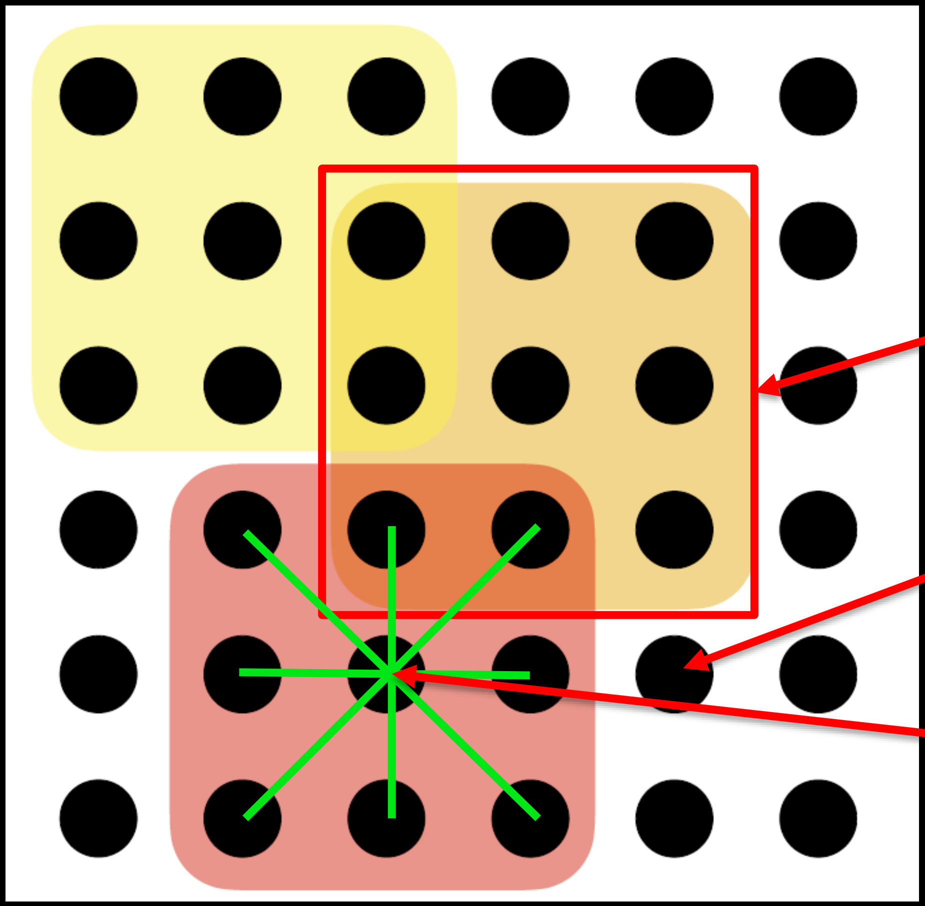
- If you can turn your object solid – you can classify it
  - Can only handle closed surfaces
- Pre-aligning an object to a canonical orientation is ill-posed
  - Only partially rotationally invariant
- Massive loss of resolution when using this volumetric representation

Extrinsic vs. intrinsic

Extrinsic   Intrinsic

Cao, Wenming, et al. "A comprehensive survey on geometric deep learning." *IEEE Access* 8 (2020): 35929-35949.

Convolution – a conceptual heads-up



Your data – an image, a mesh, a graph or something more exotic

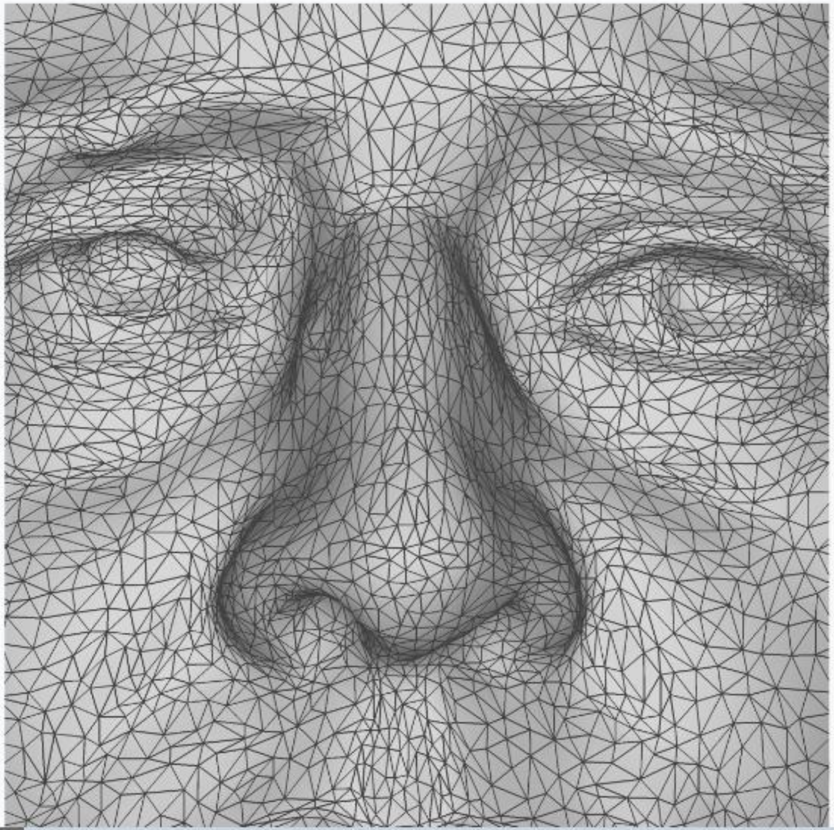
A kernel – containing (learnable) weights

For each ”node” in your data you have values

For each ”node” in your data you have a neighborhood that should be ”covered” by the kernel

When poll is active, respond at **pollev.com/rasmuspaulse538**

How many edge neighbours does an edge have in its 1-ring neighbohood?



1

2

3

4

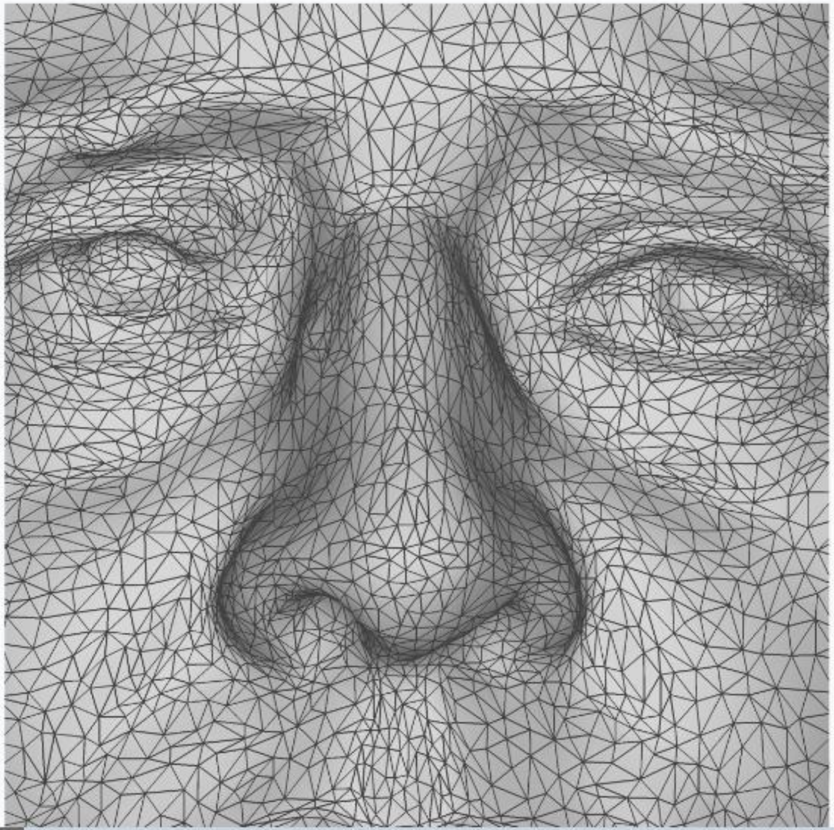
5

6

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

When poll is active, respond at **pollev.com/rasmuspaulse538**

How many edge neighbours does an edge have in its 1-ring neighbohood?

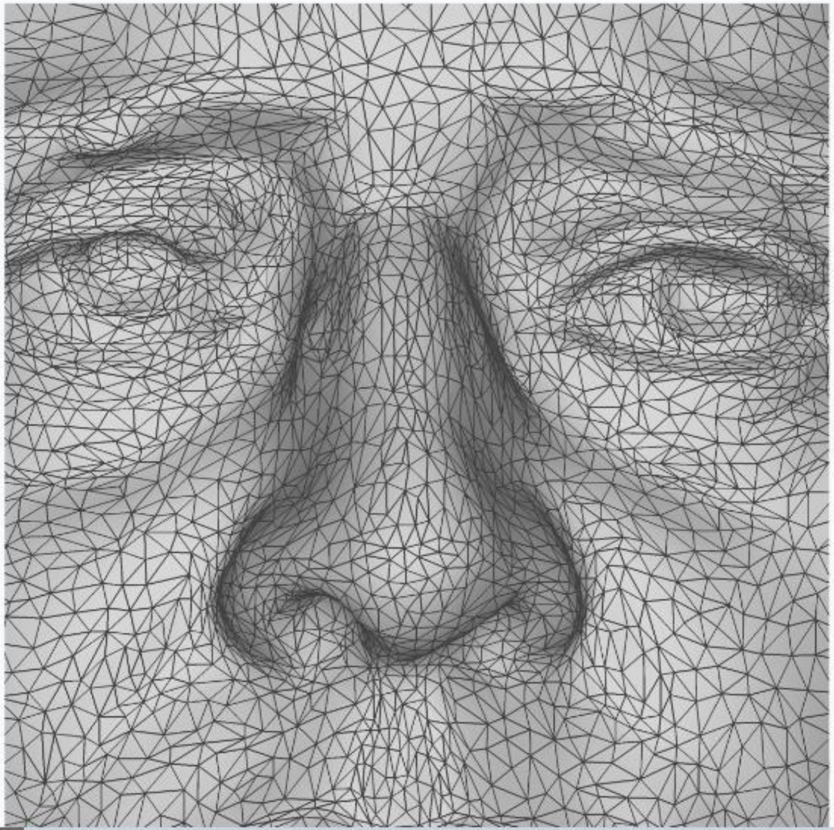


- 1
- 2
- 3
- 4
- 5
- 6

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

When poll is active, respond at **pollev.com/rasmuspaulse538**

How many edge neighbours does an edge have in its 1-ring neighbohood?



1

2

3

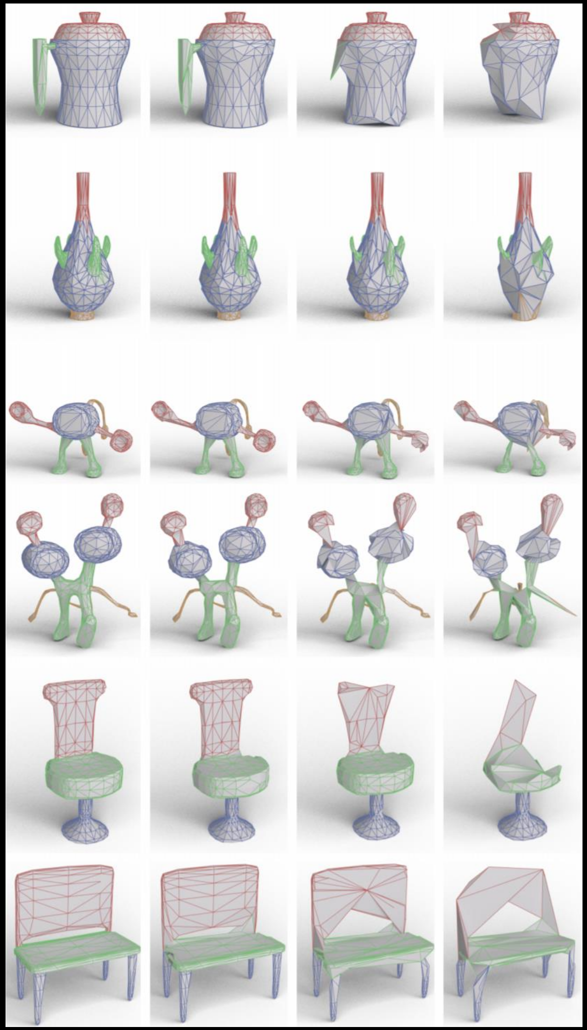
4

5

6

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Hybrid methods based on mesh operations for convolutions and pooling

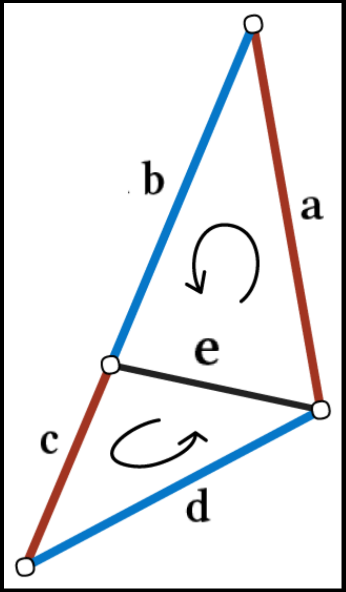


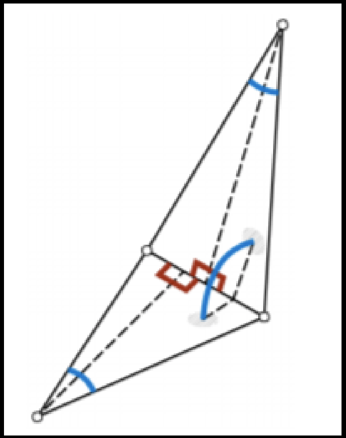
- MeshCNN used for semantic segmentation of 3D objects.
- The labelling is done per edge
- To the left the result of the segmentation
- Second, third and fourth row show simplified/reduced/pooled meshes

https://ranahanocka.github.io/MeshCNN/

Hanocka, Rana, et al. "Meshcnn: a network with an edge." *ACM Transactions on Graphics (TOG)* 38.4 (2019): 1-12.

MeshCNN – node (edge) data (features)

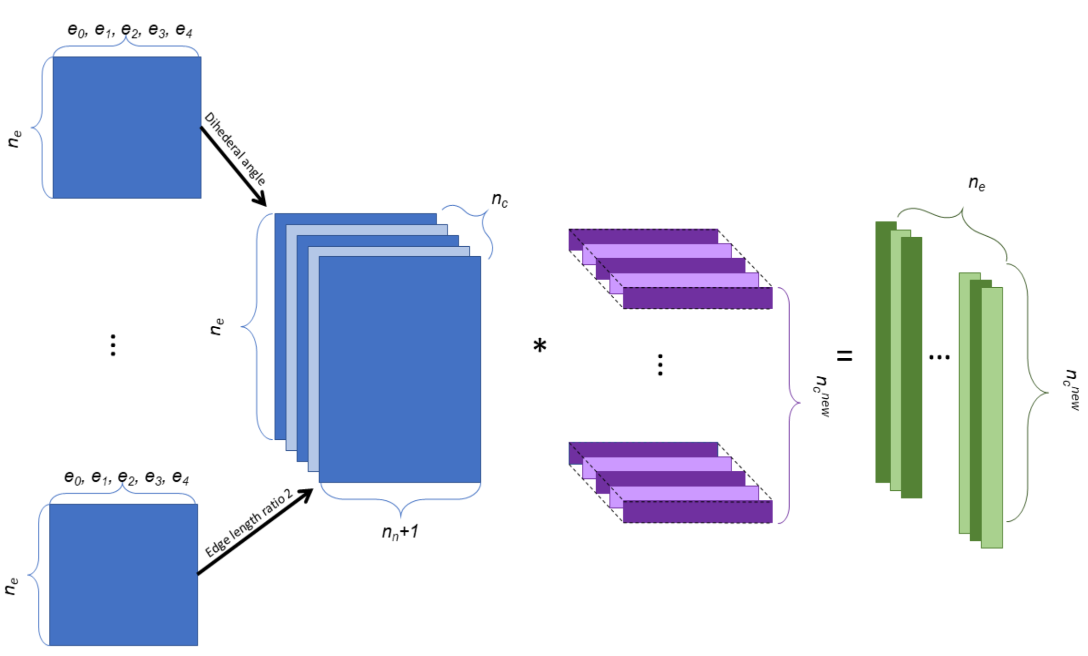




- Five features per edge:
  - The dihedral angle
  - The two inner angles
  - The two edge-length ratios
- Neighborhood of edge *e*
- *Invariant to translation, scaling and rotation*

$$(e_1, e_2, e_3, e_4) = (a + c, b + d, |a - c|, |b - d|)$$

MeshCNN – convolutions

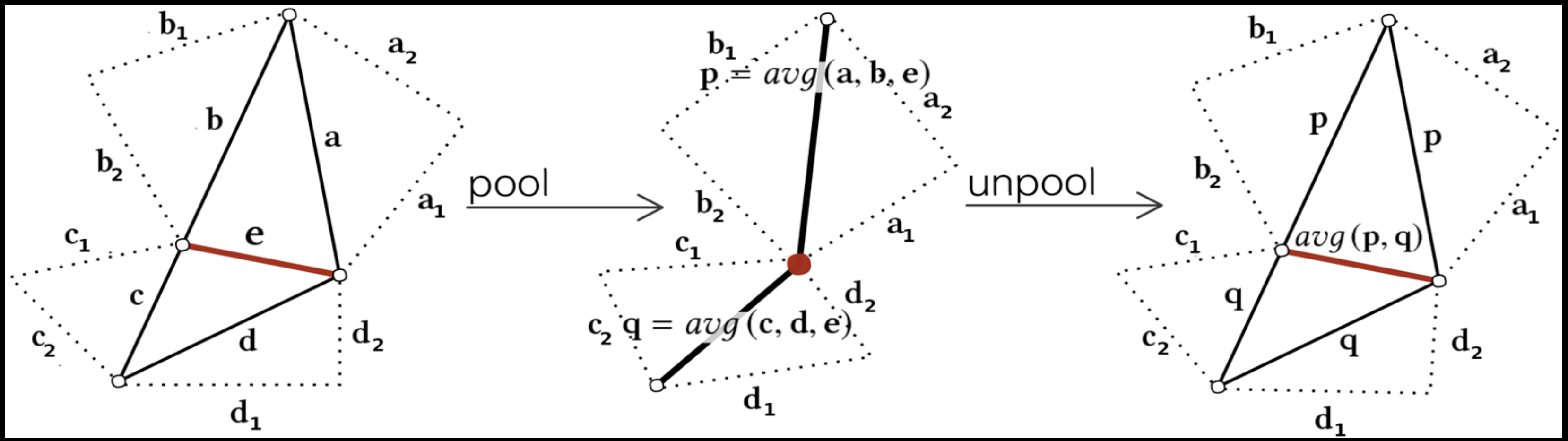


- Symmetric features on 1-ring neighbors
- Normal features for edge itself, e_0
- 1 x 5 standard 2D convolutions

$$(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4) = (\mathbf{a}+\mathbf{c}, \mathbf{b}+\mathbf{d}, |\mathbf{a}-\mathbf{c}|, |\mathbf{b}-\mathbf{d}|)$$

MeshCNN – pooling / unpooling

- The edge with the feature vector of lowest magnitude is collapsed – similar to standard mesh decimation
- Five edges → Two edges
- Bookkeeping matrix **G** (size #edge x #edge)



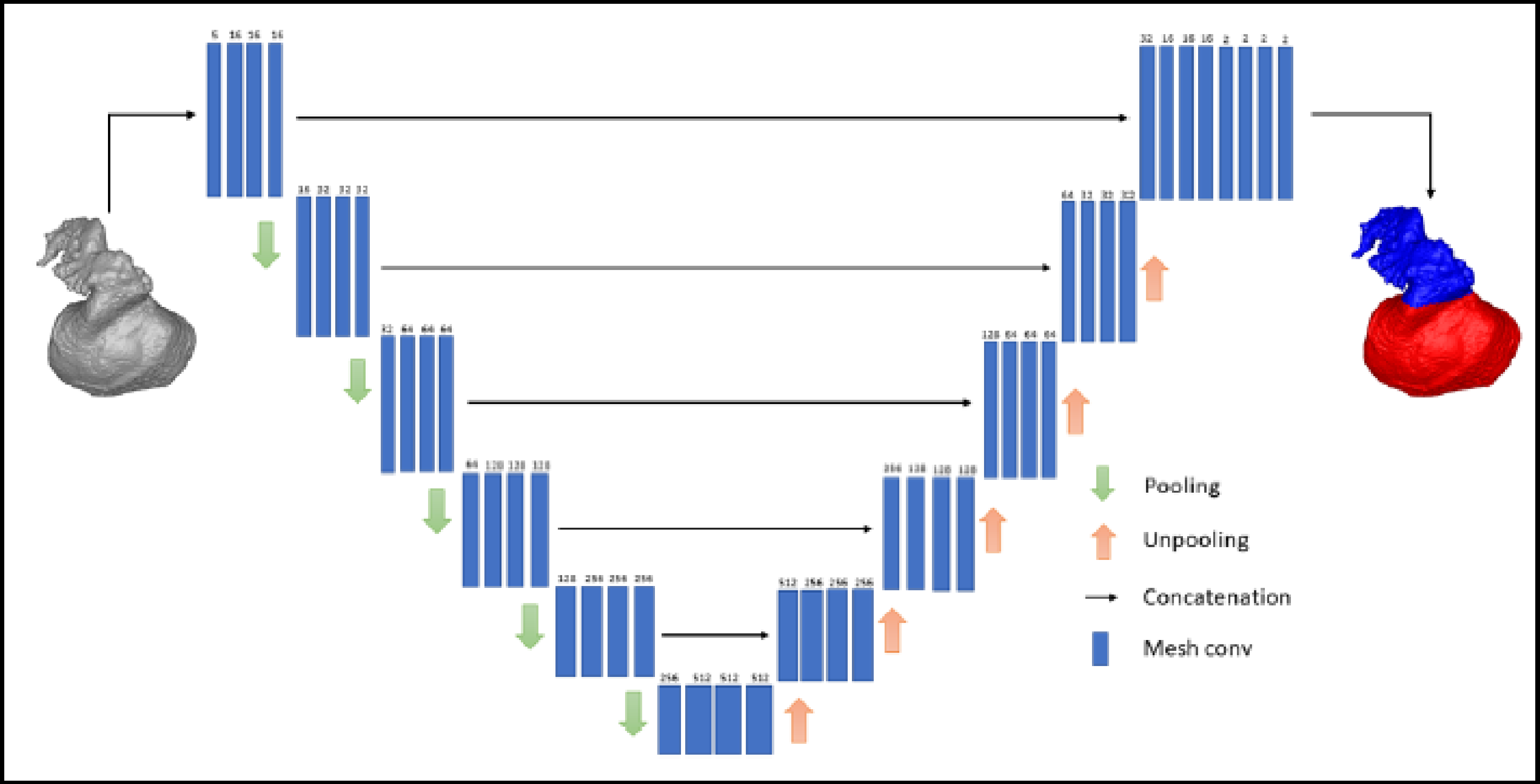
MeshCNN – network architectures

Segmentation (Down)
ResConv $f_{in} \times 32$
MeshPool $\longrightarrow$ 1800
ResConv 32×64
MeshPool $\longrightarrow$ 1350
ResConv 64×128
MeshPool $\longrightarrow$ 600
ResConv 128×256

Symmetric up- and down path

MeshCNN with U-net architecture

Based on BSc work of Bjørn Marius Schreblowski Hansen & Mathias Micheelsen Lowes

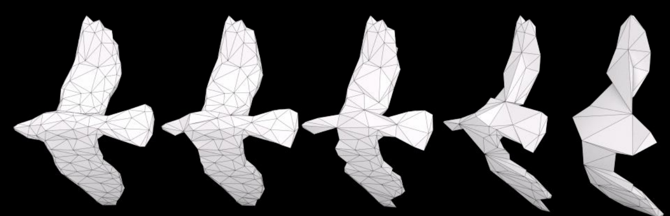


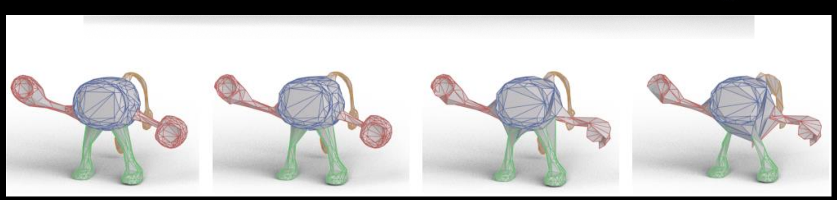
MeshCNN - results

Classification SHREC

Method	Split 16	Split 10
MeshCNN	**98.6** %	**91.0**%
GWCNN	96.6%	90.3%
GI	96.6%	88.6%
SN	48.4%	52.7%
SG	70.8%	62.6%

} [Ezuz et al. 2017]

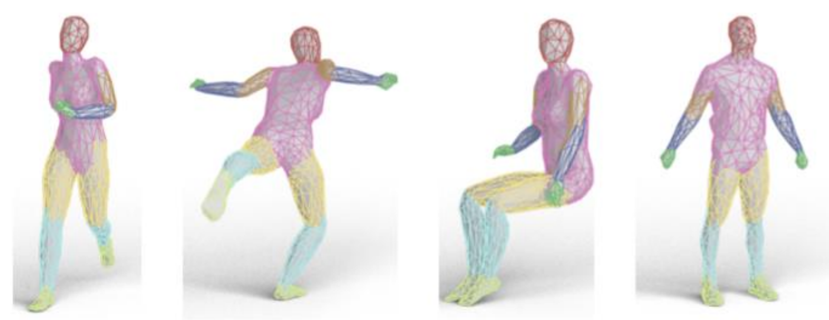




Human Body Segmentation

Method	# Features	Accuracy
MeshCNN	5	**92.30**%
SNGC	3	91.02%
Toric Cover	26	88.00%
PointNet++	3	90.77%
DynGraphCNN	3	89.72%
GCNN	64	86.40%
MDGCNN	64	89.47%

} [2018]

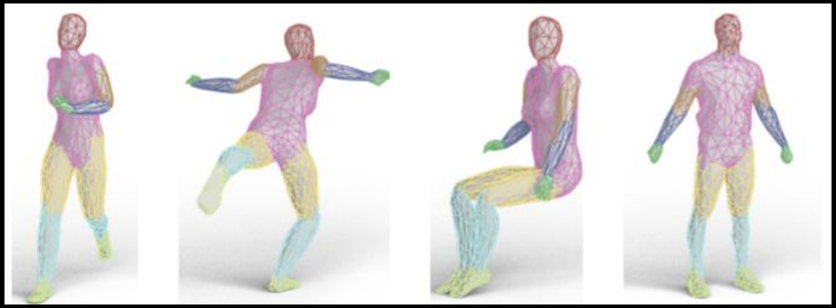


https://ranahanocka.github.io/MeshCNN/

Hanocka, Rana, et al. "Meshcnn: a network with an edge." *ACM Transactions on Graphics (TOG)* 38.4 (2019): 1-12.

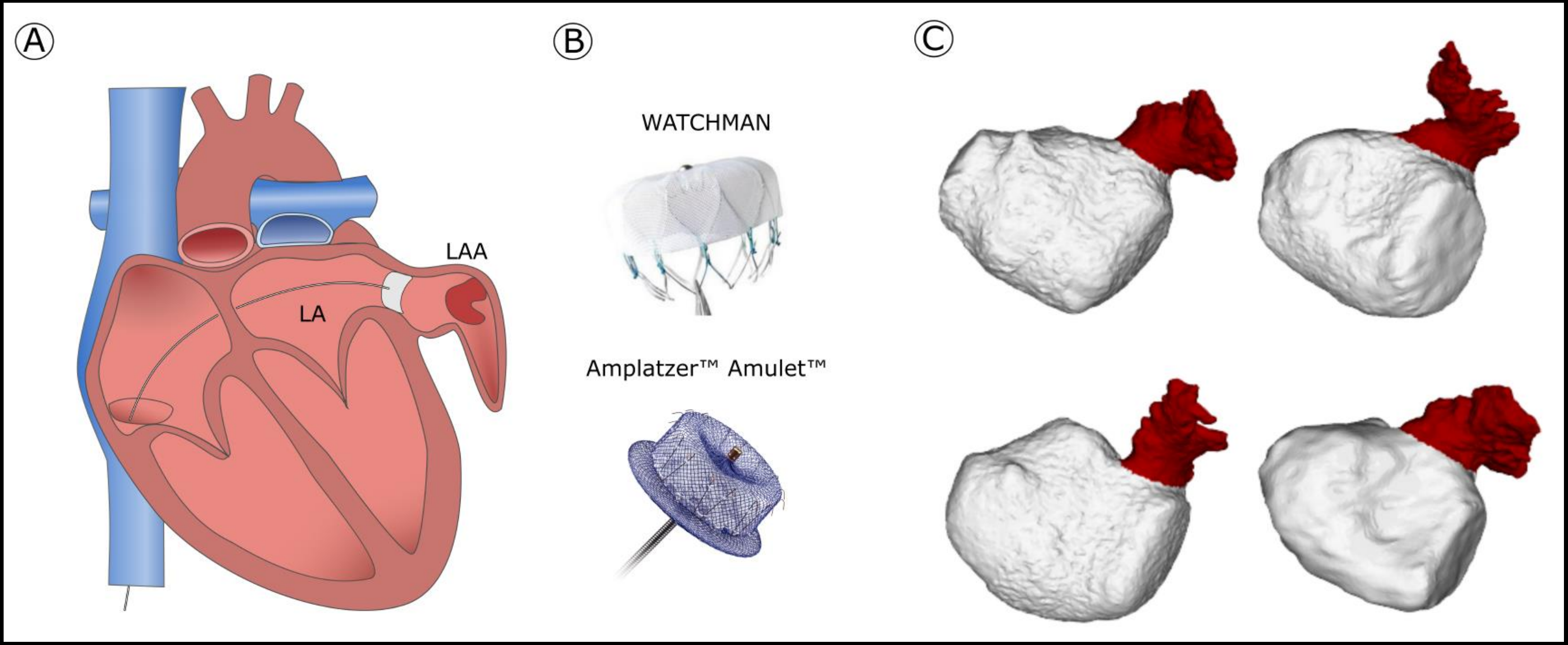
MeshCNN - observations

- Achieved impressive segmentation results on standard datasets
- Invariant to rotation, scaling and translation
- Limited to small meshes with a few hundred edges
  - Due to N^2 memory foot prints (in matrix **G**)
- Vulnerable to mesh topology and surfaces being manifold
  - Can create non-manifold surfaces during pooling



Sparse MeshCNN with attentation - paper in review

Based on BSc work of Bjørn Marius Schreblowski Hansen & Mathias Micheelsen Lowes

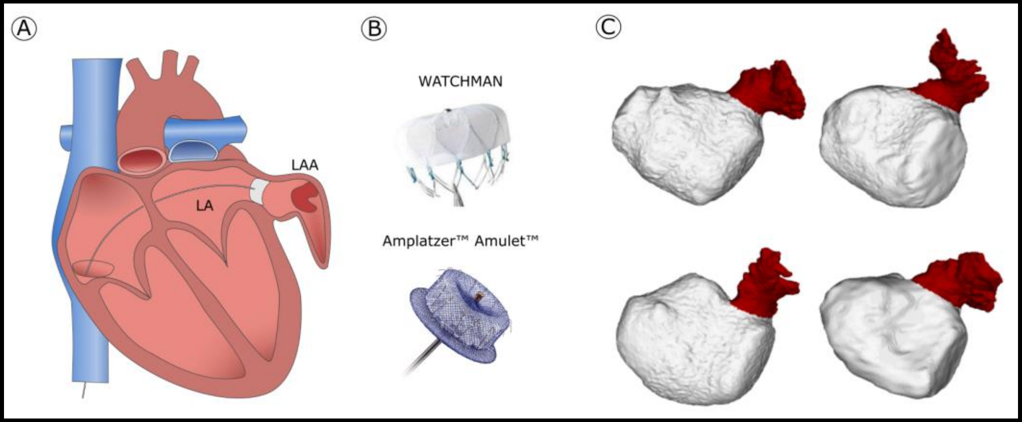


Prediction of intersection between the left atrium and the left atrial appendage in the human heart. For simulation of surgical device insertion.

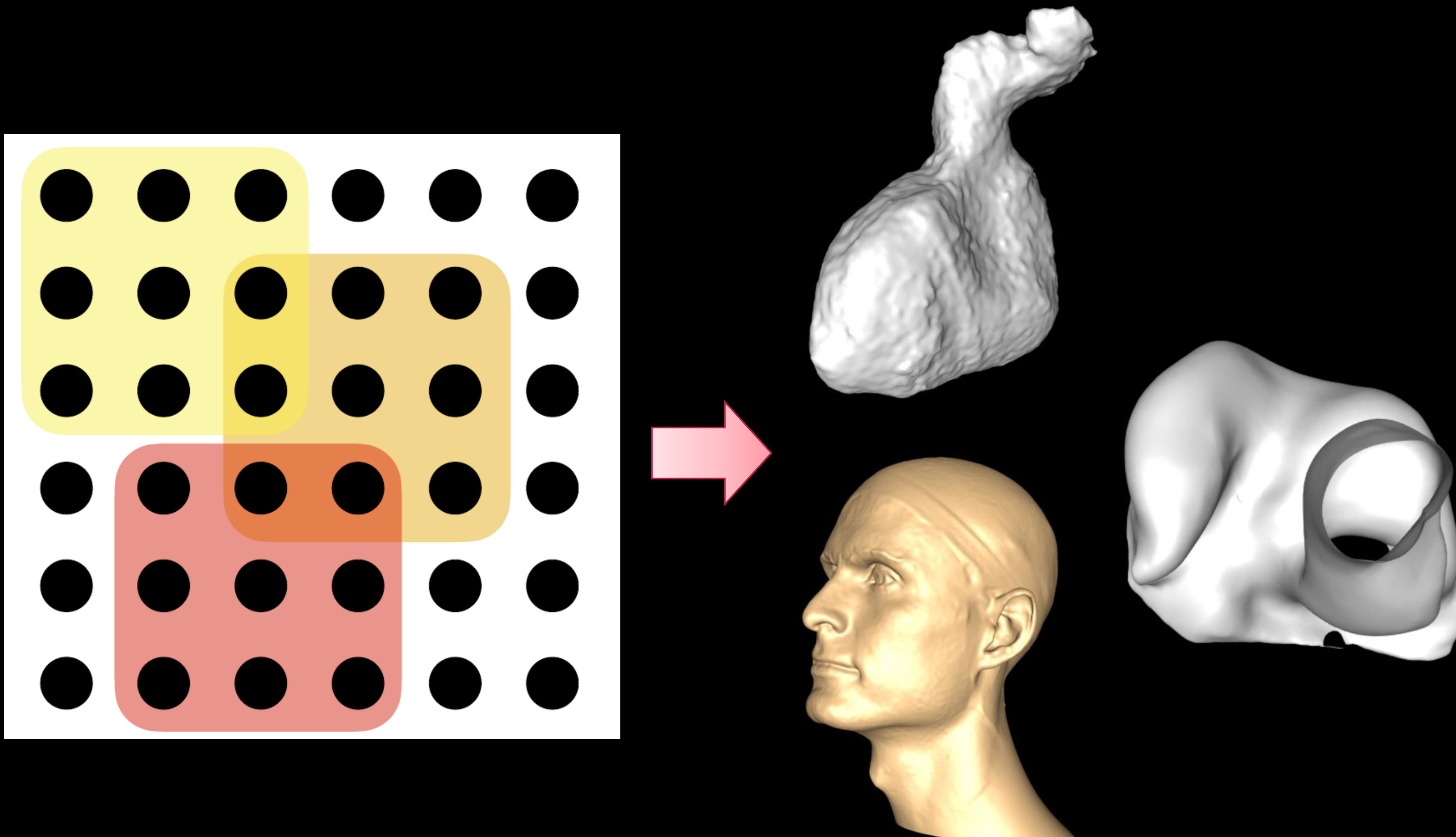
Sparse MeshCNN

- In MeshCNN
  - The matrix **G** is of size n_e^2
  - Scales quadratically with mesh size

- In Sparse MeshCNN
  - The matrix **G** is sparse
  - Can operate on larger meshes



Methods based on convolutions on meshes



PyTorch geometric

latest

Search docs

Convolutional Layers

`MessagePassing`	Base class for creating message passing layers of the form
`GCNConv`	The graph convolutional operator from the "Semi-supervised Classification with Graph Convolutional Networks" paper
`ChebConv`	The chebyshev spectral graph convolutional operator from the "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering" paper
`SAGEConv`	The GraphSAGE operator from the "Inductive Representation Learning on Large Graphs" paper
`GraphConv`	The graph neural network operator from the "Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks" paper
`GravNetConv`	The GravNet operator from the "Learning Representations of Irregular Particle-detector Geometry with Distance-weighted Graph Networks" paper, where the graph is dynamically constructed using nearest neighbors.
`GatedGraphConv`	The gated graph convolution operator from the "Gated Graph Sequence Neural Networks" paper
`ResGatedGraphConv`	The residual gated graph convolutional operator from the "Residual Gated Graph ConvNets" paper
`GATConv`	The graph attentional operator from the "Graph Attention Networks" paper
`GATv2Conv`	The GATv2 operator from the "How Attentive are Graph Attention Networks?" paper, which fixes the static attention problem of the standard `GATConv` layer: since the linear layers in the standard GAT are applied right after each other, the ranking of attended nodes is unconditioned on the query node.
`TransformerConv`	The graph transformer operator from the "Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification" paper
`AGNNConv`	The graph attentional propagation layer from the "Attention-based Graph Neural Network for Semi-Supervised Learning" paper
`TAGConv`	The topology adaptive graph convolutional networks operator from the "Topology Adaptive Graph Convolutional Networks" paper
`GINConv`	The graph isomorphism operator from the "How Powerful are Graph Neural Networks?" paper
`GINEConv`	The modified `GINConv` operator from the "Strategies for Pre-training Graph Neural Networks" paper
`ARMAConv`	The ARMA graph convolutional operator from the "Graph Neural Networks with Convolutional ARMA Filters" paper
`SGConv`	The simple graph convolutional operator from the "Simplifying Graph Convolutional Networks" paper
`APPNP`	The approximate personalized propagation of neural predictions layer from the "Predict then Propagate: Graph Neural Networks meet Personalized PageRank" paper

`MFConv`	The graph neural network operator from the "Convolutional Networks on Graphs for Learning Molecular Fingerprints" paper
`RGCNConv`	The relational graph convolutional operator from the "Modeling Relational Data with Graph Convolutional Networks" paper
`FastRGCNConv`	See `RGCNConv`.
`SignedConv`	The signed graph convolutional operator from the "Signed Graph Convolutional Network" paper
`DNAConv`	The dynamic neighborhood aggregation operator from the "Just Jump: Towards Dynamic Neighborhood Aggregation in Graph Neural Networks" paper
`PointNetConv`	The PointNet set layer from the "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" and "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space" papers
`PointConv`	alias of `torch_geometric.nn.conv.point_conv.PointNetConv`
`GMMConv`	The gaussian mixture model convolutional operator from the "Geometric Deep Learning on Graphs and Manifolds using Mixture Model CNNs" paper
`SplineConv`	The spline-based convolutional operator from the "SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels" paper
`NNConv`	The continuous kernel-based convolutional operator from the "Neural Message Passing for Quantum Chemistry" paper.
`ECConv`	alias of `torch_geometric.nn.conv.nn_conv.NNConv`
`CGConv`	The crystal graph convolutional operator from the "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties" paper
`EdgeConv`	The edge convolutional operator from the "Dynamic Graph CNN for Learning on Point Clouds" paper
`DynamicEdgeConv`	The dynamic edge convolutional operator from the "Dynamic Graph CNN for Learning on Point Clouds" paper (see `torch_geometric.nn.conv.EdgeConv`), where the graph is dynamically constructed using nearest neighbors in the feature space.
`XConv`	The convolutional operator on $\mathcal{X}$-transformed points from the "PointCNN: Convolution On X-Transformed Points" paper
`PPFConv`	The PPFNet operator from the "PPFNet: Global Context Aware Local Features for Robust 3D Point Matching" paper
`FeaStConv`	The (translation-invariant) feature-steered convolutional operator from the "FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis" paper
`HypergraphConv`	The hypergraph convolutional operator from the "Hypergraph Convolution and Hypergraph Attention" paper
`LEConv`	The local extremum graph neural network operator from the "ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations" paper, which finds the importance of nodes with respect to their neighbors using the difference operator:
`PNAConv`	The Principal Neighbourhood Aggregation graph convolution operator from the "Principal Neighbourhood Aggregation for Graph Nets" paper
`ClusterGCNConv`	The ClusterGCN graph convolutional operator from the "Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks" paper
`GENConv`	The GENeralized Graph Convolution (GENConv) from the "DeeperGCN: All You Need to Train Deeper GCNs" paper.
`GCN2Conv`	The graph convolutional operator with initial residual connections and identity mapping (GCNII) from the "Simple and Deep Graph Convolutional Networks" paper
`PANConv`	The path integral based convolutional operator from the "Path Integral Based Convolution and Pooling for Graph Neural Networks" paper
`WLConv`	The Weisfeiler Lehman operator from the "A Reduction of a Graph to a Canonical Form and an Algebra Arising During this Reduction" paper, which iteratively refines node colorings:
`FiLMConv`	The FiLM graph convolutional operator from the "GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation" paper
`SuperGATConv`	The self-supervised graph attentional operator from the "How to Find Your Friendly Neighborhood: Graph Attention Design with Self-Supervision" paper

PyTorch geometric

latest

Search docs

NOTES

- Installation
- Introduction by Example
- Creating Message Passing Networks
- Creating Your Own Datasets
- Advanced Mini-Batching
- Memory-Efficient Aggregations
- TorchScript Support
- GNN Cheatsheet
- Colab Notebooks
- External Resources

PACKAGE REFERENCE

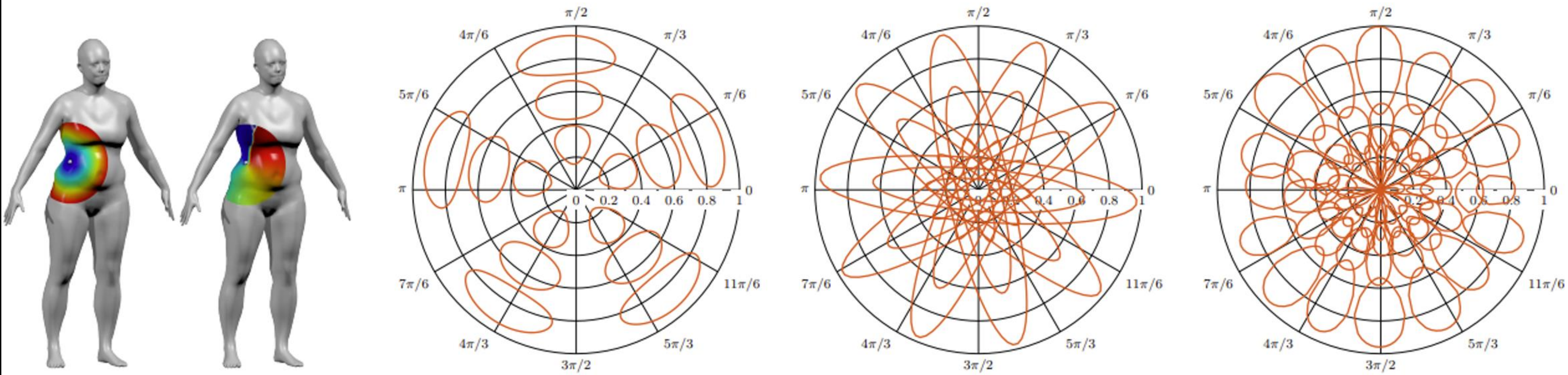
- torch_geometric
- torch_geometric.nn
  - Convolutional Layers
  - Dense Convolutional Layers
  - Normalization Layers
  - Global Pooling Layers
  - Pooling Layers
  - Dense Pooling Layers
  - Unpooling Layers
  - Models
  - Functional
  - DataParallel Layers
- torch_geometric.data
- torch_geometric.datasets
- torch_geometric.transforms
- torch_geometric.utils
- torch_geometric.io

Read the Docs v: latest

Pooling Layers

`TopKPooling`	top_k pooling operator from the "Graph U-Nets", "Towards Sparse Hierarchical Graph Classifiers" and "Understanding Attention and Generalization in Graph Neural Networks" papers
`SAGPooling`	The self-attention pooling operator from the "Self-Attention Graph Pooling" and "Understanding Attention and Generalization in Graph Neural Networks" papers
`EdgePooling`	The edge pooling operator from the "Towards Graph Pooling by Edge Contraction" and "Edge Contraction Pooling for Graph Neural Networks" papers.
`ASAPooling`	The Adaptive Structure Aware Pooling operator from the "ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations" paper.
`PANPooling`	The path integral based pooling operator from the "Path Integral Based Convolution and Pooling for Graph Neural Networks" paper.
`MemPooling`	Memory based pooling layer from "Memory-Based Graph Networks" paper, which learns a coarsened graph representation based on soft cluster assignments
`max_pool`	Pools and coarsens a graph given by the `torch_geometric.data.Data` object according to the clustering defined in `cluster`.
`avg_pool`	Pools and coarsens a graph given by the `torch_geometric.data.Data` object according to the clustering defined in `cluster`.
`max_pool_x`	Max-Pools node features according to the clustering defined in `cluster`.
`max_pool_neighbor_x`	Max pools neighboring node features, where each feature in `data.x` is replaced by the feature value with the maximum value from the central node and its neighbors.
`avg_pool_x`	Average pools node features according to the clustering defined in `cluster`.
`avg_pool_neighbor_x`	Average pools neighboring node features, where each feature in `data.x` is replaced by the average feature values from the central node and its neighbors.
`graclus`	A greedy clustering algorithm from the "Weighted Graph Cuts without Eigenvectors: A Multilevel Approach" paper of picking an unmarked vertex and matching it with one of its unmarked neighbors (that maximizes its edge weight).
`voxel_grid`	Voxel grid pooling from the, *e.g.*, Dynamic Edge-Conditioned Filters in Convolutional Networks on Graphs paper, which overlays a regular grid of user-defined size over a point cloud and clusters all points within the same voxel.
`fps`	A sampling algorithm from the "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space" paper, which iteratively samples the most distant point with regard to the rest points.
`knn`	Finds for each element in `y` the `k` nearest points in `x`.
`knn_graph`	Computes graph edges to the nearest `k` points.
`radius`	Finds for each element in `y` all points in `x` within distance `r`.
`radius_graph`	Computes graph edges to all points within a given distance.
`nearest`	Clusters points in `x` together which are nearest to a given query point in `y`.

Convolutions on meshes



Polar coordinates ρ, θ | GCNN | ACNN | MoNet

Monti et al., Geometric deep learning on graphs and manifolds using mixture model CNNs, 2017, CVPR

$$(D(x)f)(\rho,\theta) = \int_X v_{\rho,\theta}(x,x')f(x')\,dx'$$

$$(f * g)(x) = \max_{\Delta\theta\in[0,2\pi)} \int_0^{2\pi}\int_0^{\rho_{\max}} g(\rho,\theta+\Delta\theta)\,(D(x)f)(\rho,\theta)\,d\rho\,d\theta$$

$$(D_\alpha(x)f)(\theta,t) = \frac{\int_X h_{\alpha\theta,t}(x,x')f(x')\,dx'}{\int_X h_{\alpha\theta,t}(x,x')\,dx'}$$

$$(f * g)(x) = \int g(\theta,t)\,(D_\alpha(x)f)(\theta,t)\,dt\,d\theta$$

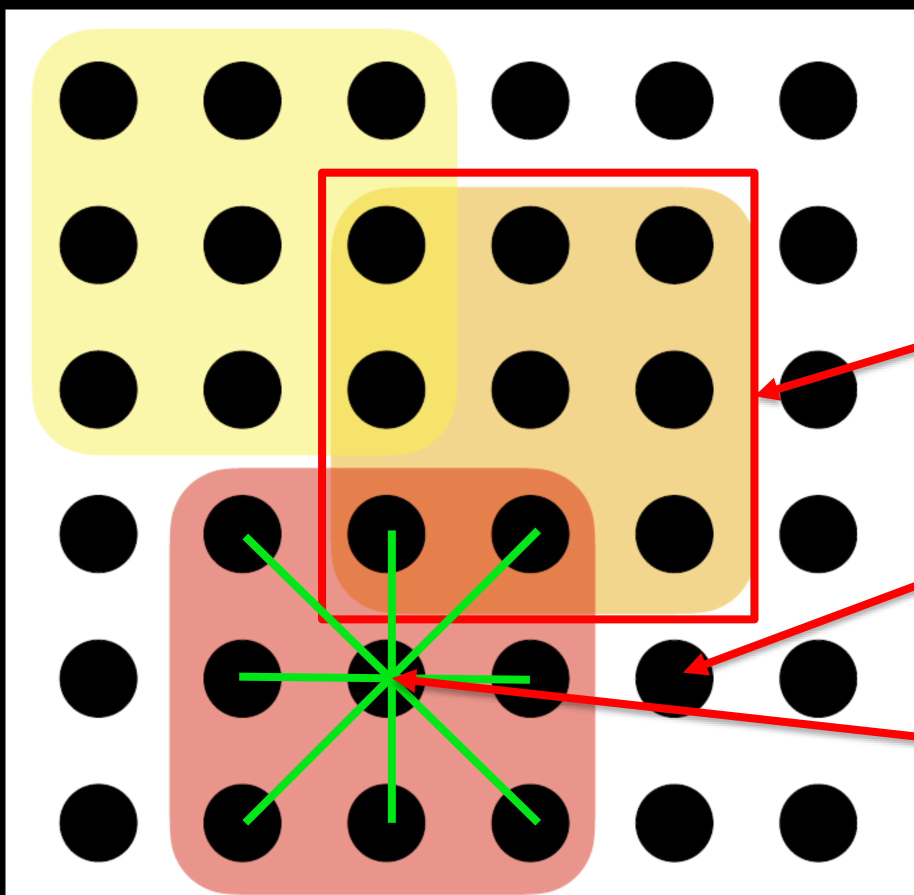
$$D_j(x)f = \sum_{x' \in N(x)} w_j(\mathbf{u}(x,x'))f(x'),$$

$$(f * g)(x) = \sum_{j=1}^{J} g_j D_j(x) f$$

Method	Pseudo-coordinates	$\mathbf{u}(x,y)$	Weight function $w_j(\mathbf{u}), j = 1,\ldots,J$
CNN [23]	Local Euclidean	$\mathbf{x}(x,y) = \mathbf{x}(y) - \mathbf{x}(x)$	$\delta(\mathbf{u} - \bar{\mathbf{u}}_j)$
GCNN [26]	Local polar geodesic	$\rho(x,y), \theta(x,y)$	$\exp(-\frac{1}{2}(\mathbf{u}-\bar{\mathbf{u}}_j)^\top \begin{pmatrix} \bar{\sigma}_\rho^2 & \\ & \bar{\sigma}_\theta^2 \end{pmatrix}^{-1}(\mathbf{u}-\bar{\mathbf{u}}_j))$
ACNN [7]	Local polar geodesic	$\rho(x,y), \theta(x,y)$	$\exp(-\frac{1}{2}\mathbf{u}^\top \mathbf{R}_{\bar{\theta}_j} \begin{pmatrix} \bar{\alpha} & \\ & 1 \end{pmatrix} \mathbf{R}_{\bar{\theta}_j}^\top \mathbf{u})$
GCN [21]	Vertex degree	$\deg(x), \deg(y)$	$\left(1 - \lvert 1 - \frac{1}{\sqrt{u_1}} \rvert\right)\left(1 - \lvert 1 - \frac{1}{\sqrt{u_2}} \rvert\right)$
DCNN [3]	Transition probability in r hops	$p^0(x,y),\ldots,p^{r-1}(x,y)$	$\mathrm{id}(u_j)$

Cao, Wenming, et al. "A comprehensive survey on geometric deep learning." *IEEE Access* 8 (2020): 35929-35949.

Convolution – a conceptual heads-up



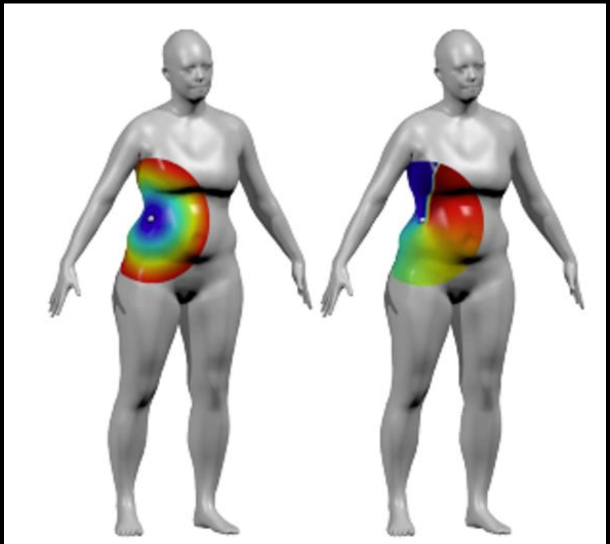
Your data – an image, a mesh, a graph or something more exotic

A kernel – containing (learnable) weights

For each ”node” in your data you have values

For each ”node” in your data you have a neighborhood that should be ”covered” by the kernel

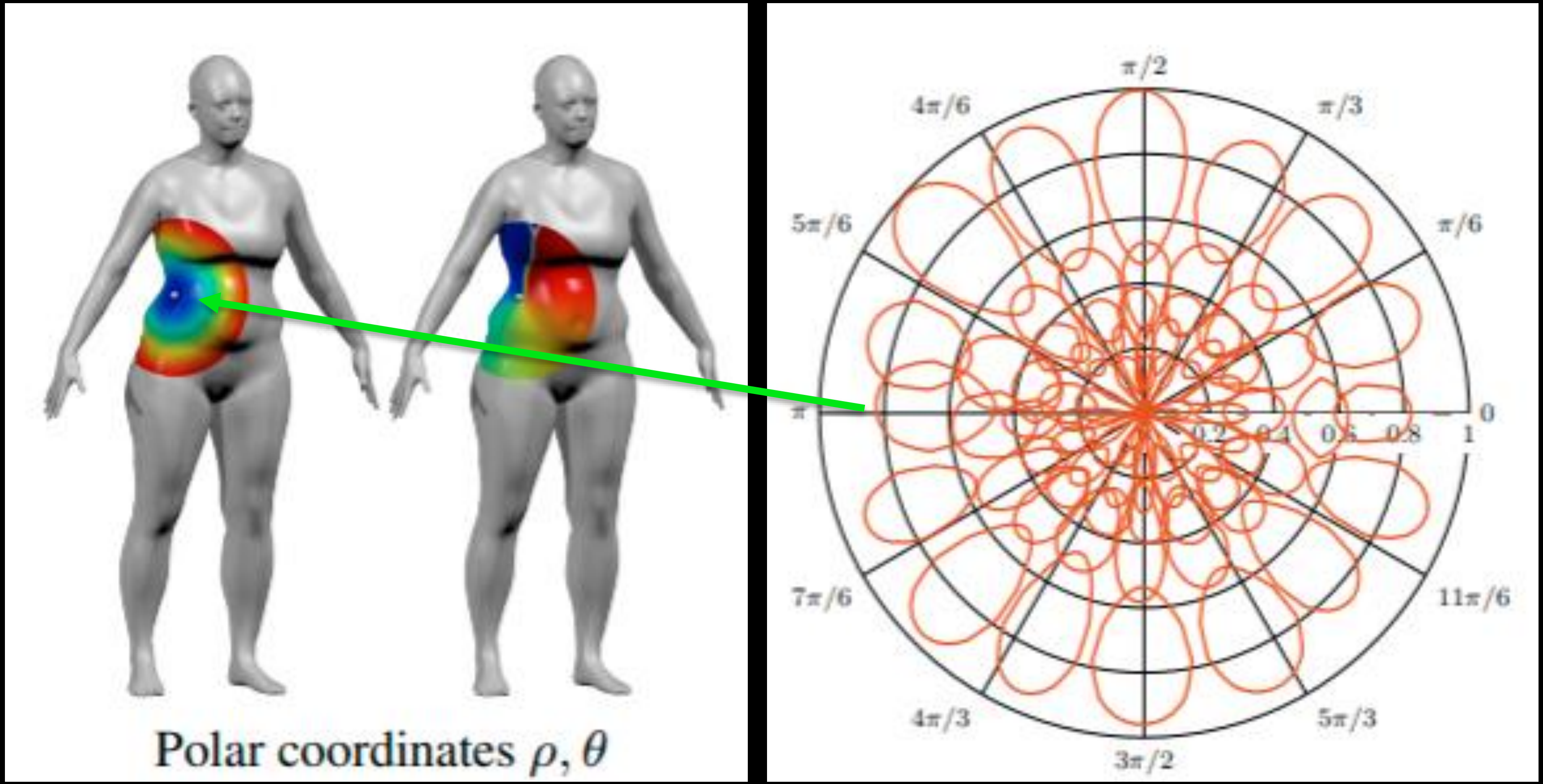
Convolutions on meshes



- Main differences between approaches
  - How is a node neighborhood defined / computed
  - What values are used per node
  - How are the weights in the convolutions defined
  - How are we dealing with kernel rotational invariance?

Cao, Wenming, et al. "A comprehensive survey on geometric deep learning." *IEEE Access* 8 (2020): 35929-35949.

One example - MoNet



Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. *Proc.CVPR*.

MoNet – vertex features

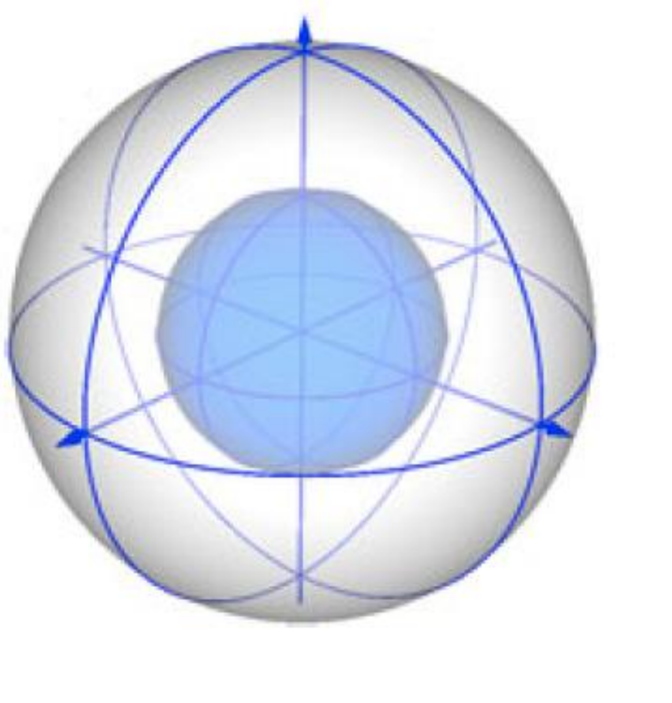


Fig. 4. Signature structure for SHOT

- Vertex features should represent local geometry
- Local shape signature
  - Histogram of local normal vectors
  - 544 dimensional vector (per vertex)

Tombari et al. "Unique signatures of histograms for local surface description." *European conference on computer vision*. 2010.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. *Proc.CVPR*.

MoNet – vertex features.
Bam! Back to classical shape matching

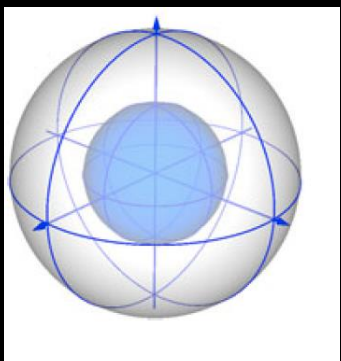
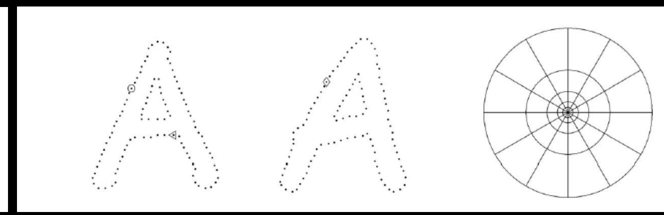


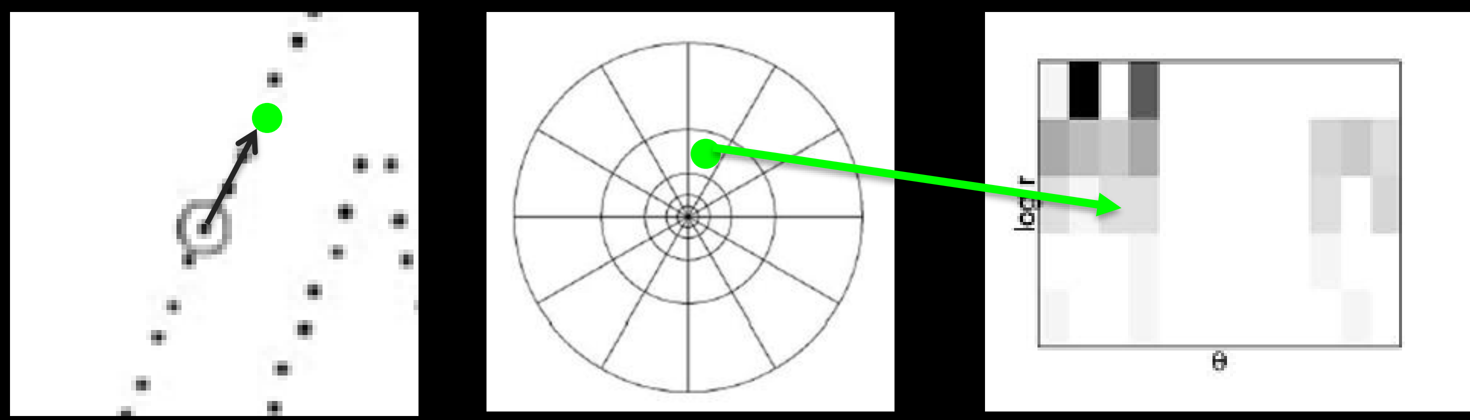
Fig. 4. Signature structure for SHOT

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 24, APRIL 2002

Shape Matching and Object Recognition Using Shape Contexts

Serge Belongie, *Member*, *IEEE*, Jitendra Malik, *Member*, *IEEE*, and Jan Puzicha





The local shape descriptor used in MoNet is similar to 3D extensions of shape contexts – and comes with its own choices, strengths and weaknesses.

Tombari et al. "Unique signatures of histograms for local surface description." *European conference on computer vision*. 2010.

Local reference frame

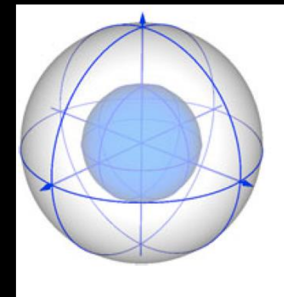
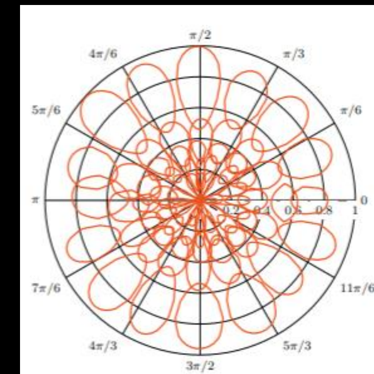
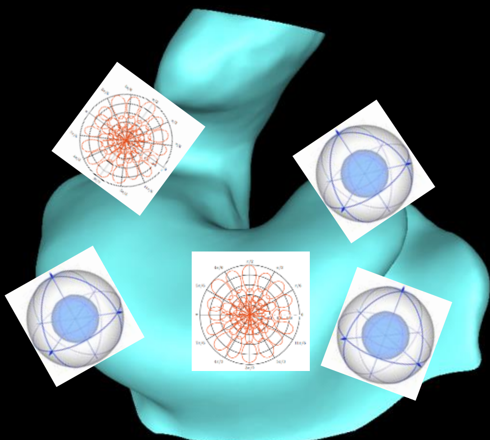
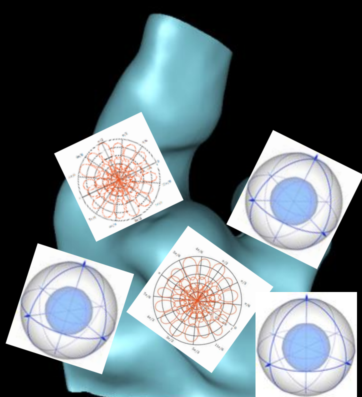


Fig. 4. Signature structure for SHOT

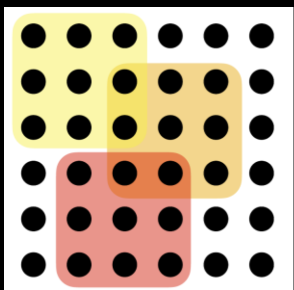




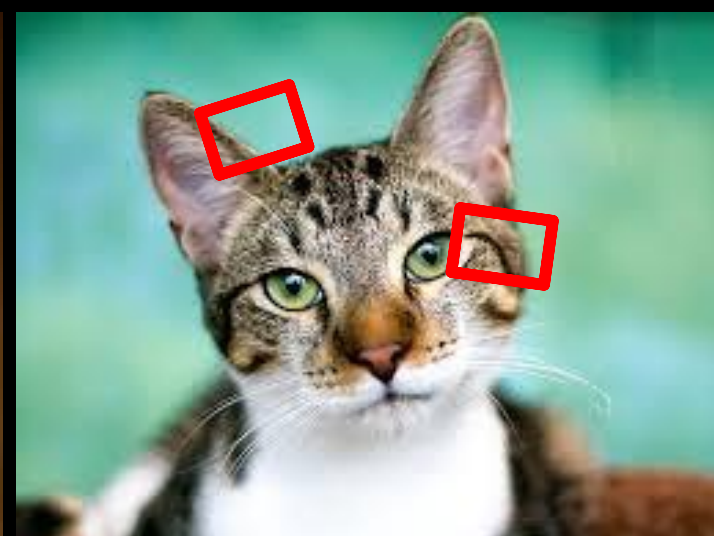


- The local reference frame is the per-vertex coordinate system
- Determines the orientation of SHOT feature extractor
- Might determine the orientation of the local convolution patch
  - Unless convolution is taken as the maximum over all rotations (around the normal) of the patch

Inconsistent local reference frame



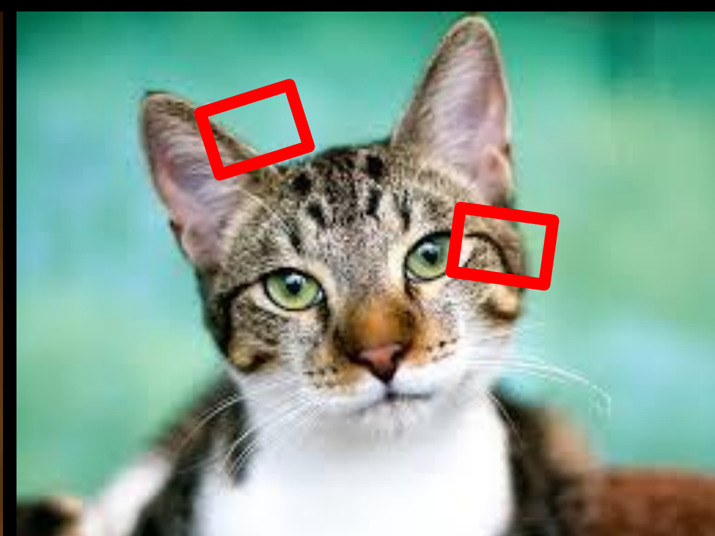
- Imagine that you had no general orientation of your 2D image
- For each pixel, your convolution kernel has an arbitrary orientation
- That is the default situation with 3D meshes



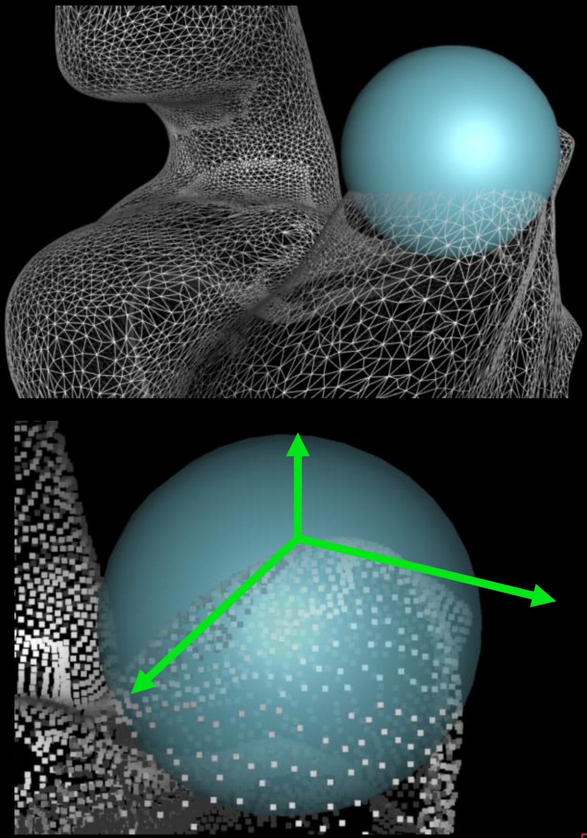
Inconsistent local reference frame

- One approach:
    - Compute local reference frame (coordinate system) following the gradient in the image
    - Convolutions would be following gradients – maybe good – maybe bad

- Another approach: Rotate kernel and take the maximum output…very expensive



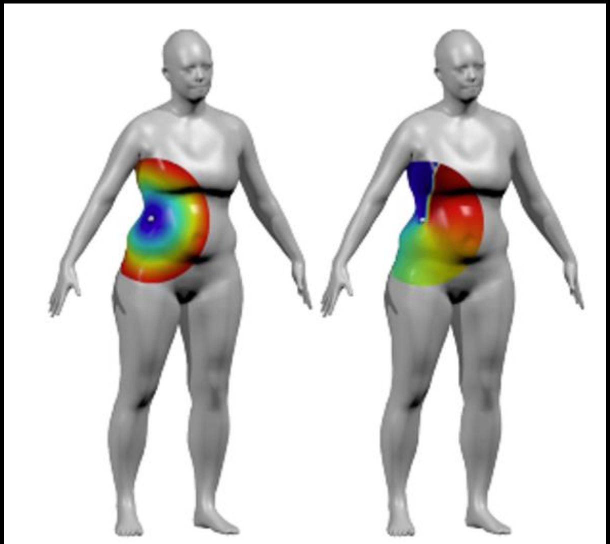
Computing a Local reference frame
- using a 3D equivalent of gradients/curvature



- Sample points in a local neighborhood
- Do eigenvector decomposition
- 3 Eigenvectors
  - One is the normal (smallest eigenvalue)
  - Two follow the surface
- Normally inconsistent and ambiguous
- Reference below claims to have solved it
- Used in MoNet (as far as I understand the paper)

Tombari et al. "Unique signatures of histograms for local surface description." *European conference on computer vision*. 2010.

MoNet and similar methods – observations



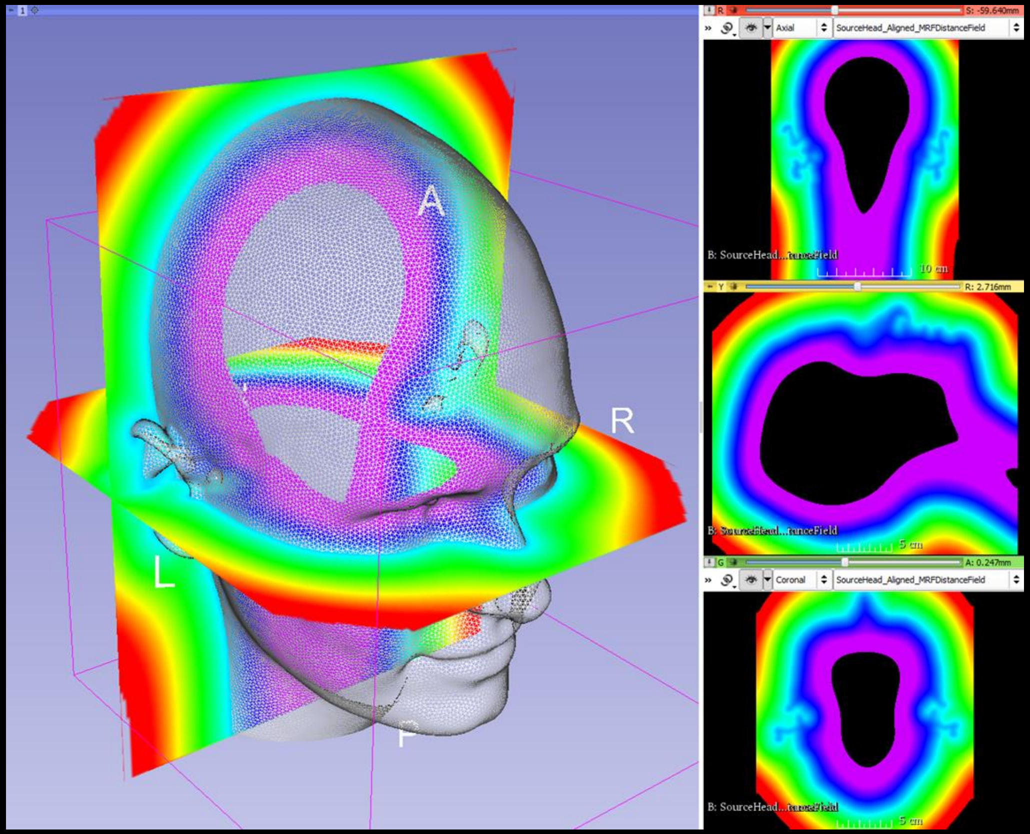
- Choice of local features to represent geometry
- Are they dependent on a consistent local reference frame?
- Topological constraints?

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. *Proc.CVPR*.

A note on spectral methods

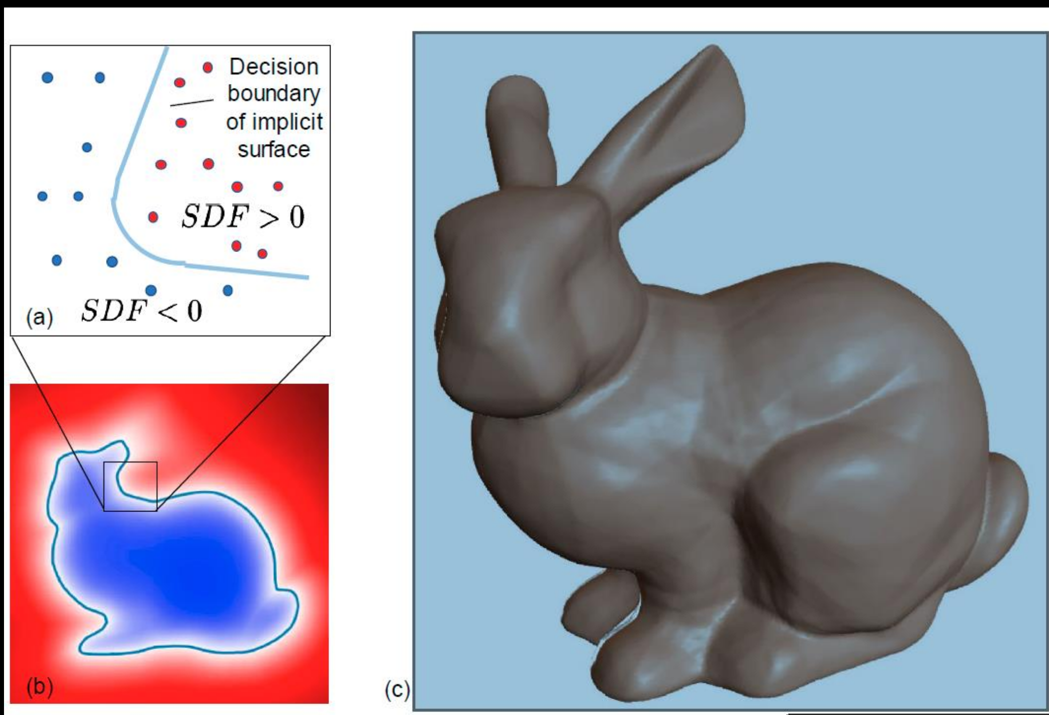
- There is a strong relation between Fourier analysis and convolutions for 1D and 2D signals
- This can be replicated on 3D meshes and is also related to the mesh Laplacian
- Quite a lot of spectral methods have been published
- It seems that the are loosing popularity and they are beyond the scope and time of this presentation
- Some comments can be found in
  - Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. Proc.CVPR.

Deep learning with implicit functions
The signed distance function

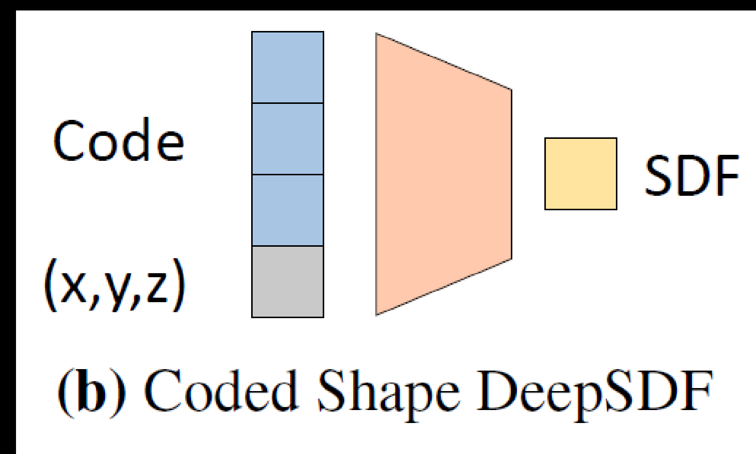


- Voxel grid – each voxel contains a scalar value
- Carries information about the shape in the entire field

DeepSDF



$$f_\theta(\boldsymbol{x}) \approx SDF(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega\,.$$

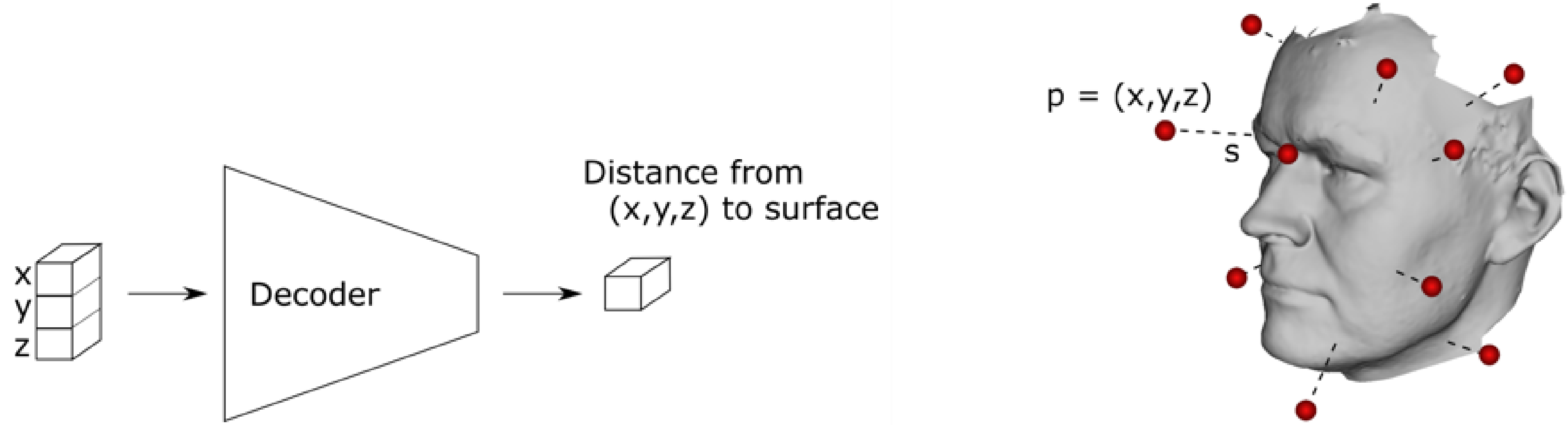


Park, Jeong Joon, et al. "Deepsdf: Learning continuous signed distance functions for shape representation." *Proc. Computer Vision and Pattern Recognition*. 2019.

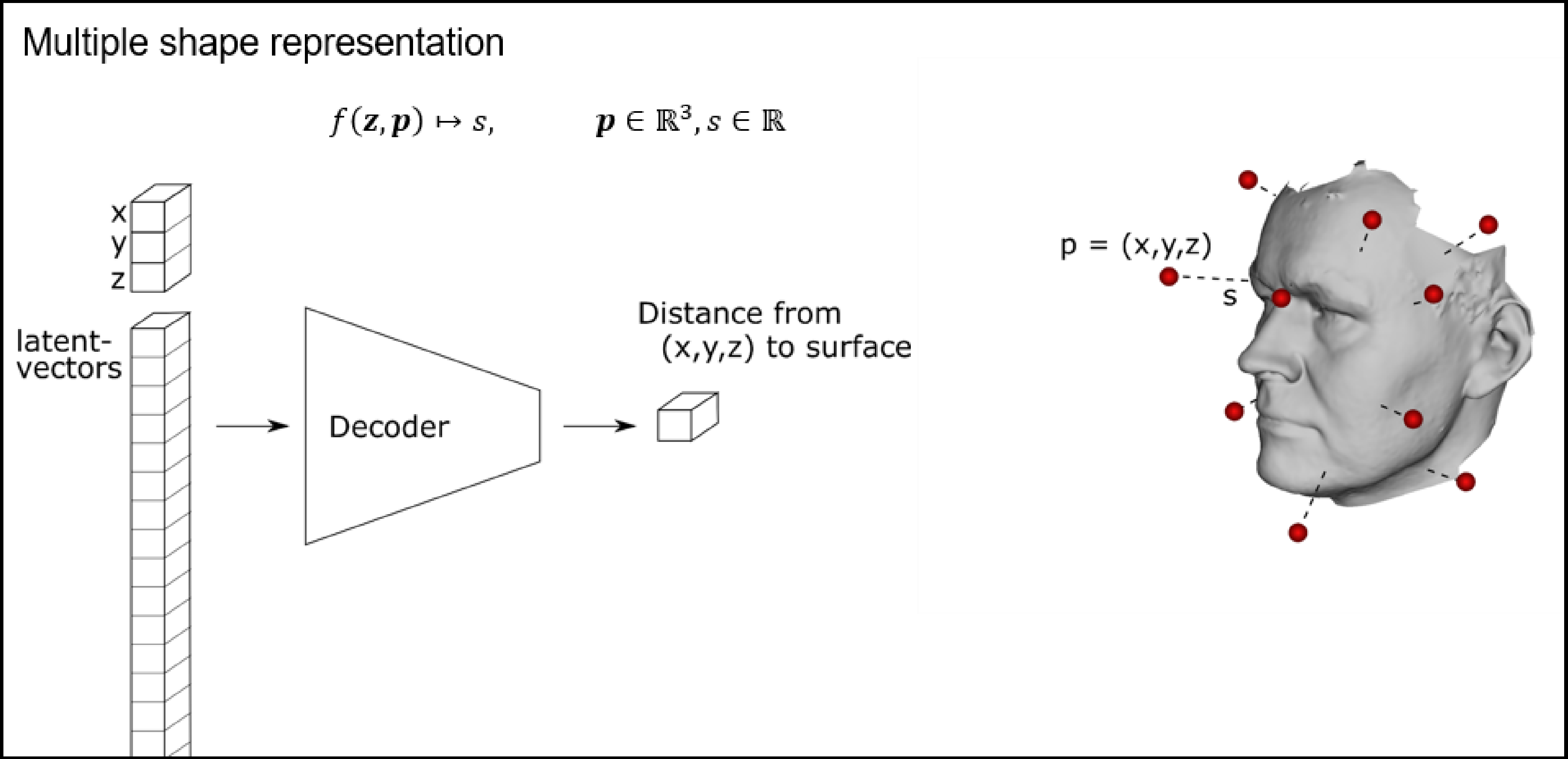
DeepSDF – single shape representation

Single shape representation

$$f(\boldsymbol{p}) \mapsto s, \qquad \boldsymbol{p} \in \mathbb{R}^3, s \in \mathbb{R}$$



DeepSDF – multiple shape representation



DeepSDF - Training

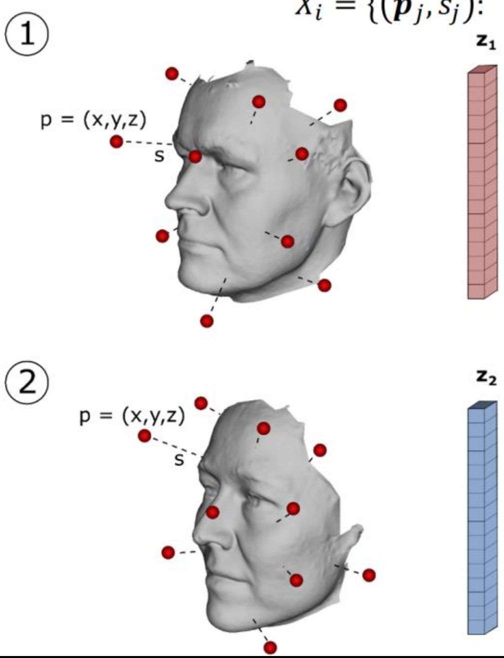
- Training:

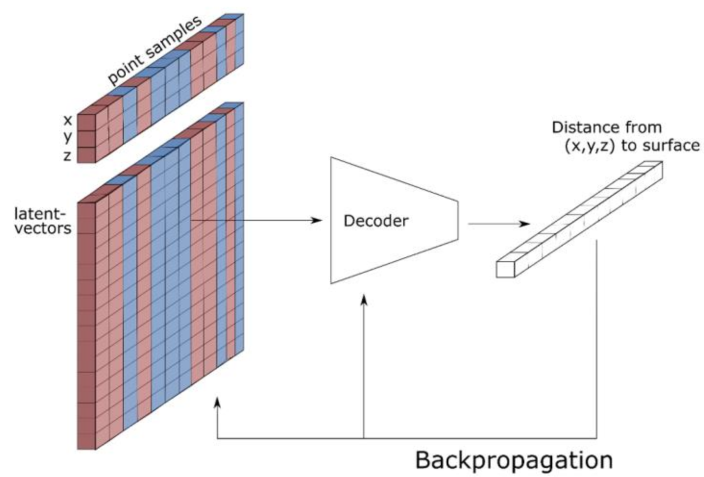
$$X_i = \{(\boldsymbol{p}_j, s_j): \quad s_j = DF^i(\boldsymbol{p}_j)\}$$

Clamped L1-distance

$$\underset{\theta,\{\mathbf{z}_i\}_{i=1}^N}{\arg\min} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \mathcal{L}(f_\theta(\mathbf{z}_i, \boldsymbol{p}_j), s_j) + \frac{1}{\sigma^2} \|\mathbf{z}_i\|_2^2 \right)$$

Regularization





Park et.al., "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation", CVPR2019

DeepSDF - results

Results:

- Reconstructing known shapes
- Reconstructing unknown shapes
- Shape completion
- Latent space shape interpolation

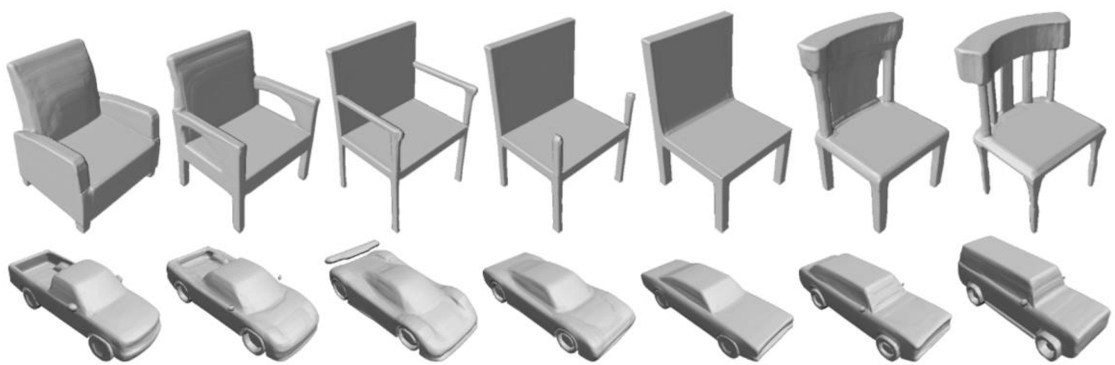
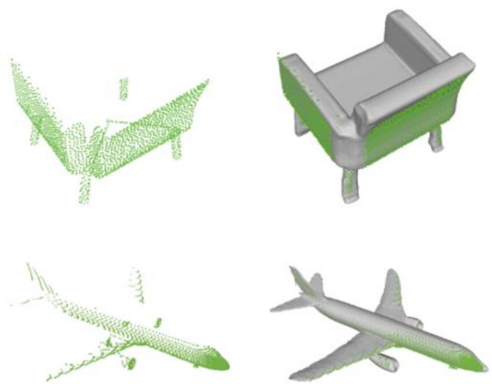


Figure 1: DeepSDF represents signed distance functions (SDFs) of shapes via latent code-conditioned feed-forward decoder networks. Above images are raycast renderings of DeepSDF interpolating between two shapes in the learned shape latent space. Best viewed digitally.

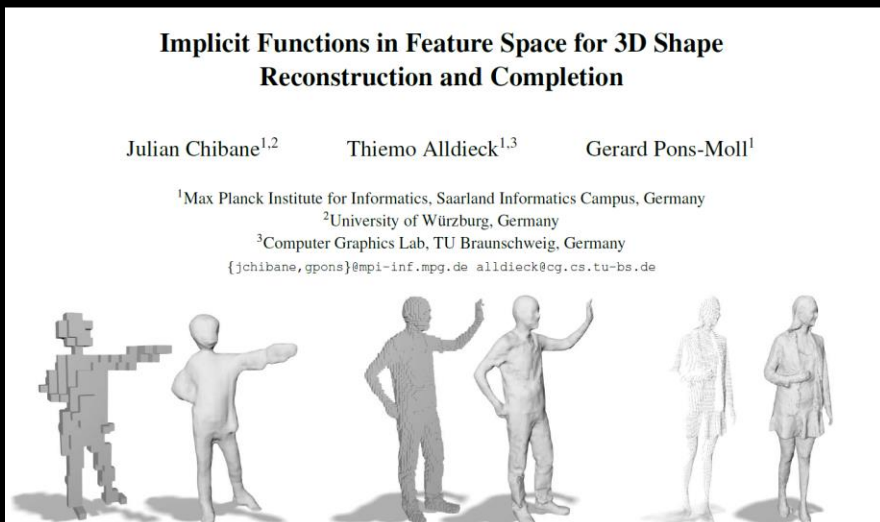


(a) Input Depth **(b)** Completion (ours)

DeepSDF - observations

- You should be able compute a signed distance to your mesh
  - Needs closed surfaces
- Not rotational invariant – unless you do heavy data augmentation
- Can do shape classification, shape synthesis and shape completion
  - Has a very usable latent space

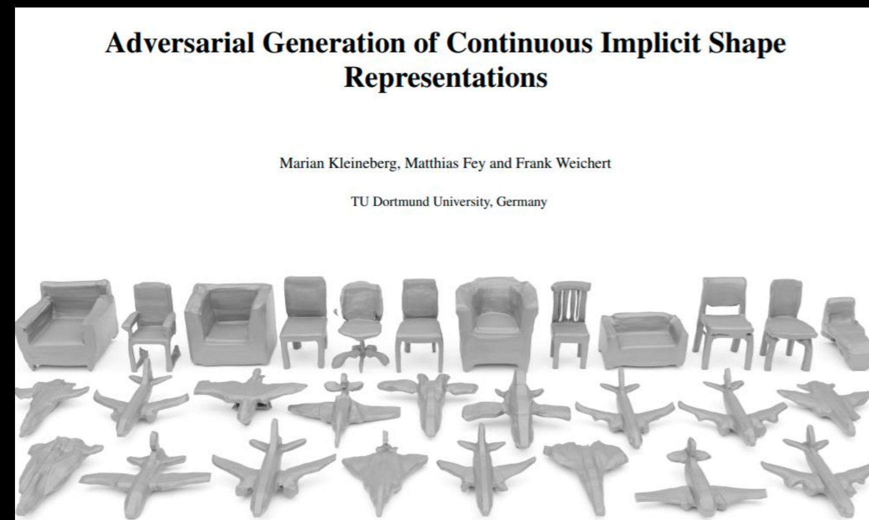
Other implicit approaches



Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion

Julian Chibane[1,2] Thiemo Alldieck[1,3] Gerard Pons-Moll[1]

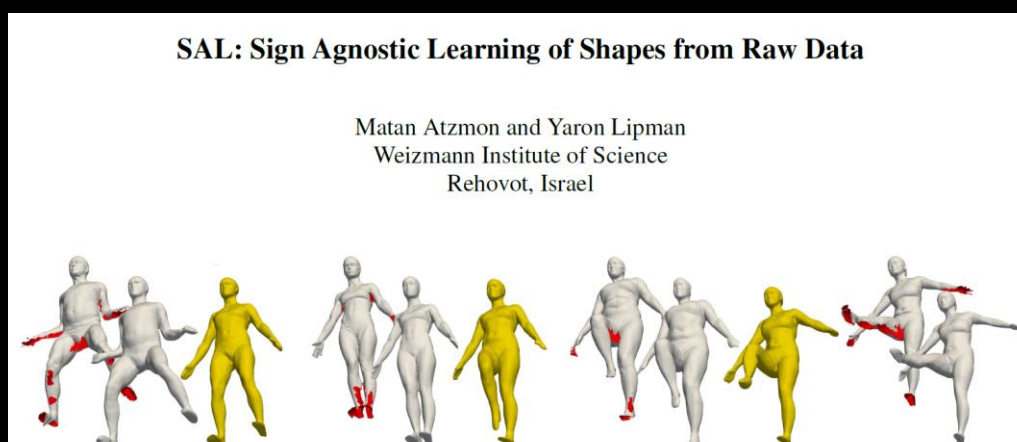
[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[2]University of Würzburg, Germany
[3]Computer Graphics Lab, TU Braunschweig, Germany
{jchibane,gpons}@mpi-inf.mpg.de alldieck@cg.cs.tu-bs.de



Adversarial Generation of Continuous Implicit Shape Representations

Marian Kleineberg, Matthias Fey and Frank Weichert

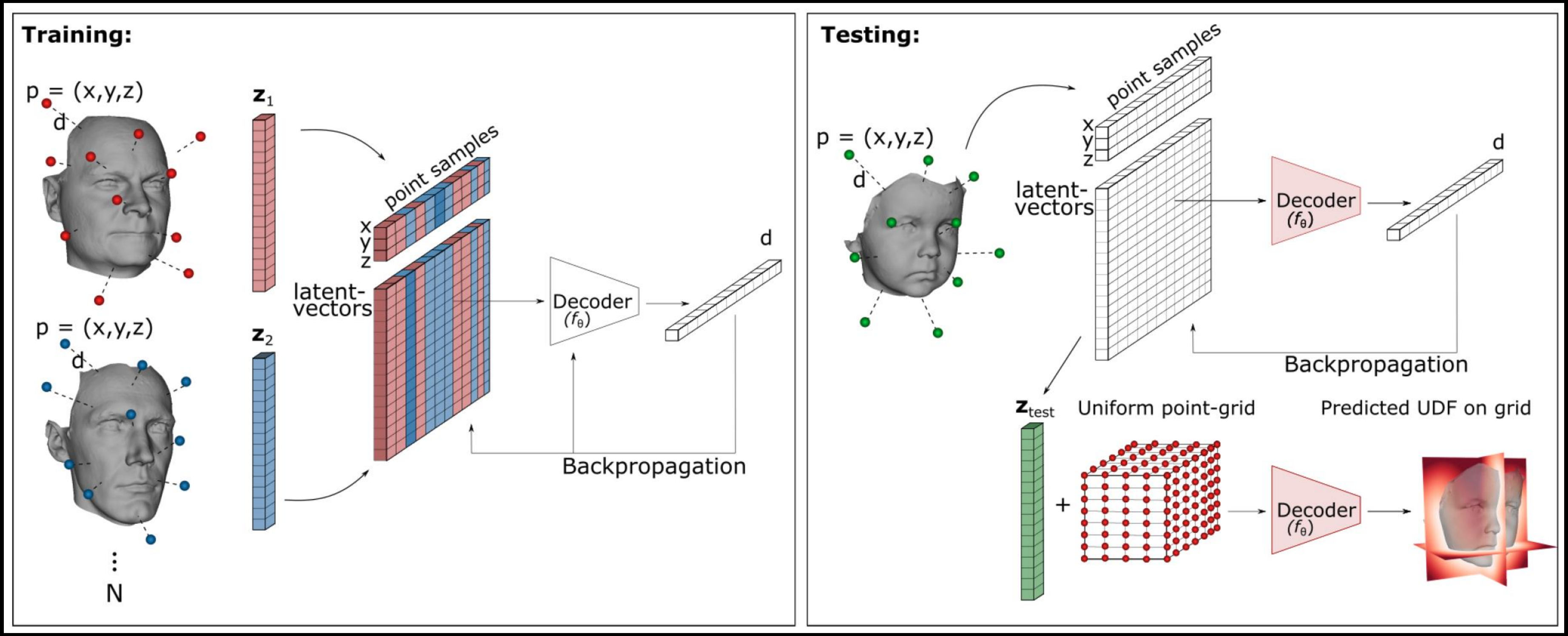
TU Dortmund University, Germany



SAL: Sign Agnostic Learning of Shapes from Raw Data

Matan Atzmon and Yaron Lipman
Weizmann Institute of Science
Rehovot, Israel

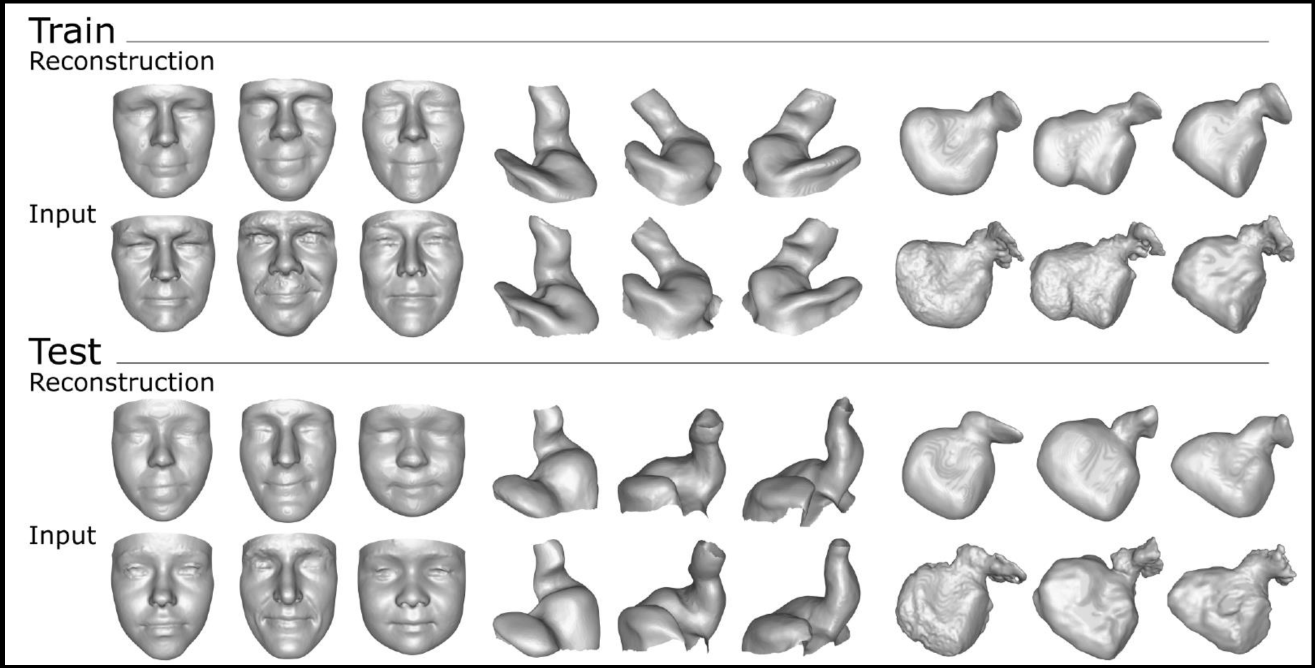
Unsigned distance fields



Implicit Neural Distance Representation for Unsupervised and Supervised Classification of Complex Anatomies. *Kristine Aavild Juhl et al.* **MICCAI 2021**

Unsigned distance fields

- Can handle arbitrary topologies
- Meshing an unsigned distance field is very tricky



Implicit Neural Distance Representation for Unsupervised and Supervised Classification of Complex Anatomies. *Kristine Aavild Juhl et al.* **MICCAI 2021**

Shape classification using unsigned distance fields

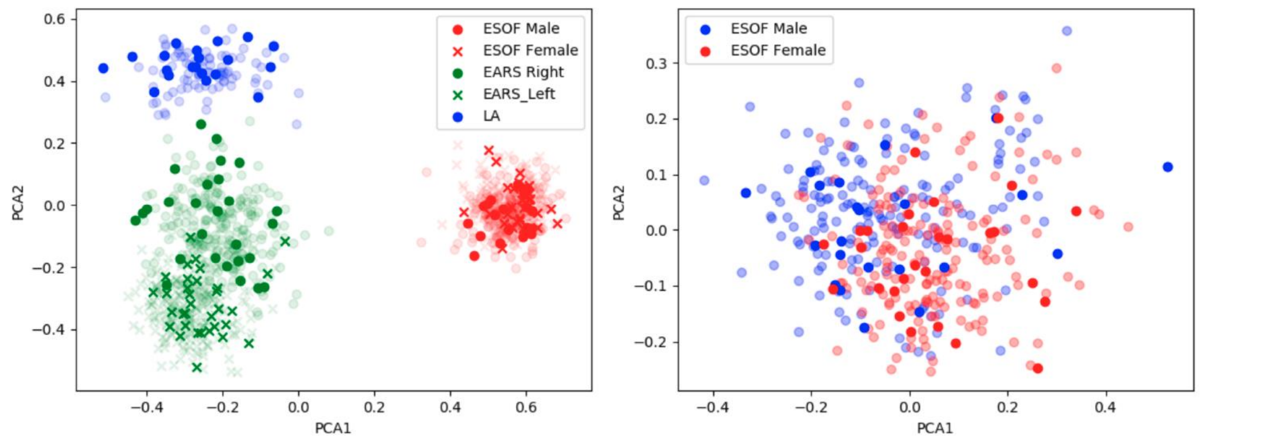
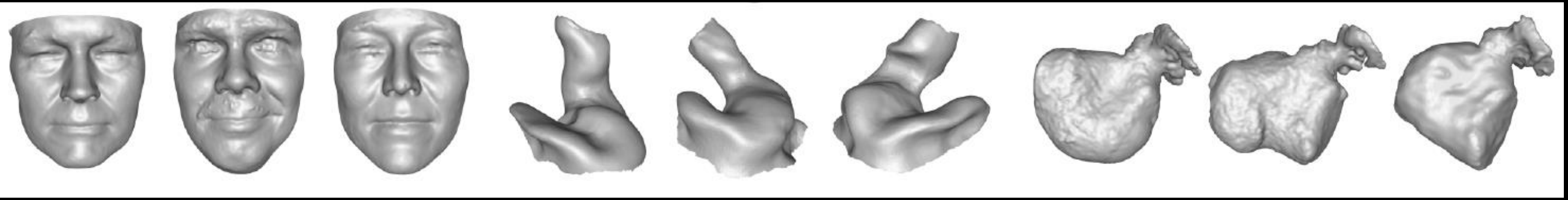


Fig. 3. Visualization of the latent space projected to 2D using principle component analysis (PCA) for the 128 dimensional latent space of the three datasets (left) and the ESOF-faces only (right). Opaque: training set, Solid: test set.



Implicit Neural Distance Representation for Unsupervised and Supervised Classification of Complex Anatomies. *Kristine Aavild Juhl et al.* **MICCAI 2021**

That's is – the tour is over!

If you have data and need a way forward

- What is the nature of my data?
  - number of samples, number of vertices, topology, cleanness, canonical orientations?
- What is my goal?
  - segmentation, classification, shape correspondence, shape completion?

- What approach fits my data and goals?
  - can it handle your data (size is a main issue)
  - can it be adapted to solve your task?
  - are there any code available
  - what are the hardware requirements (mainly GPU memory size)
  - what are the software / operating system requirements?

What if I need a more theoretical research direction?

- Find your own niche that you want to explore
- Locate an unsolved problem
- In GDL there are lots of problems
  - but also a large of number of people looking at them.
- You should have a competitive advantage
  - new idea, alternative approaches

- A new mesh convolution operator will probably have limited novelty