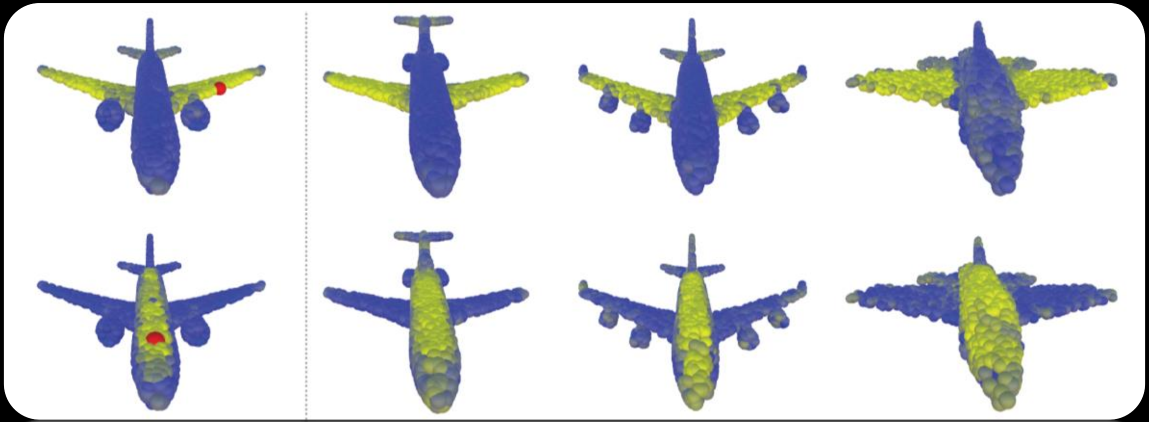
Sensible Deep Learning

for 3D Data



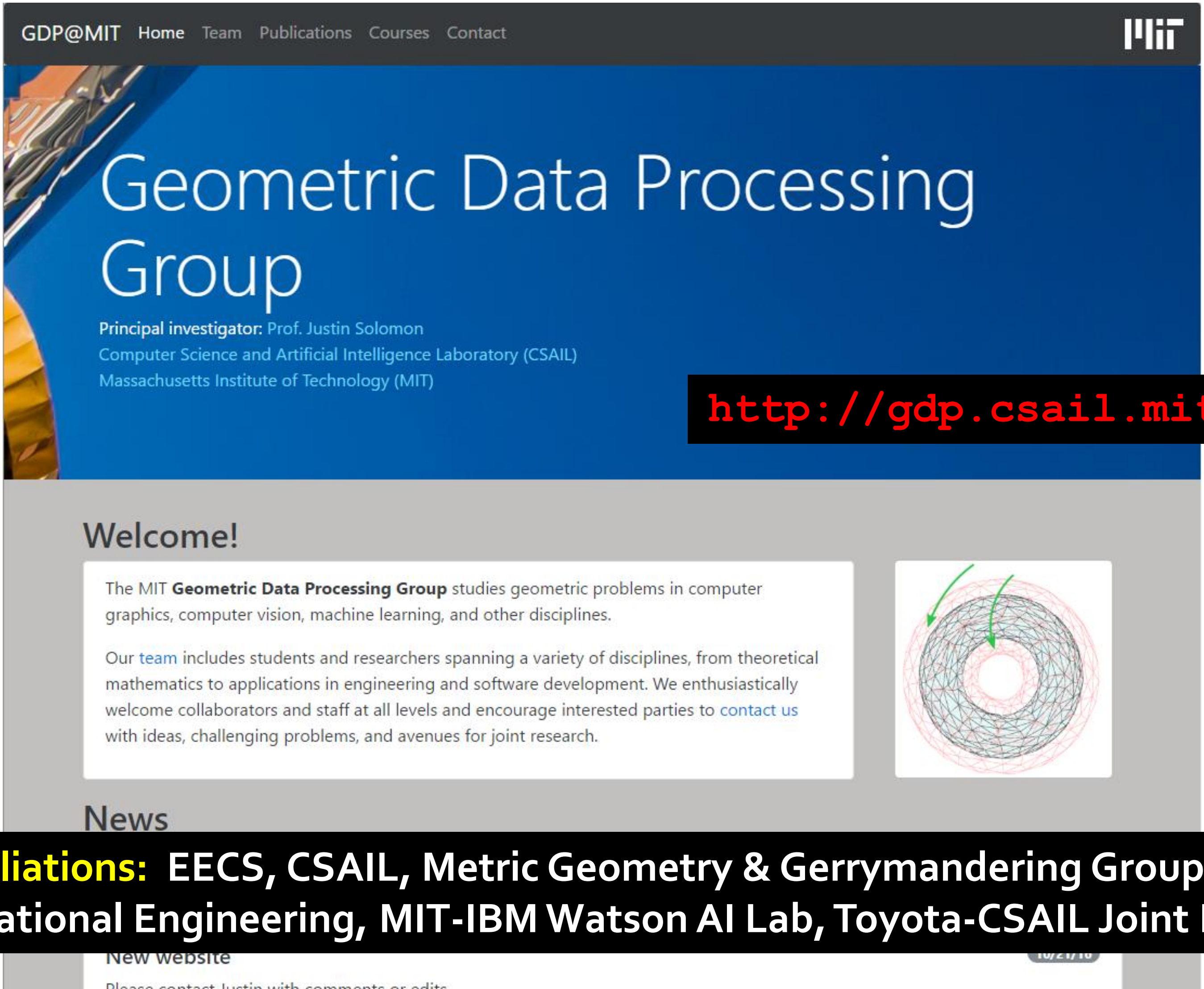
Justin Solomon

MIT EECS

NOT A PHYSICIST

But I want to learn about your geometry problems!

GDP @ MIT



Geometric Data Processing Group

Principal investigator: Prof. Justin Solomon
Computer Science and Artificial Intelligence Laboratory (CSAIL)
Massachusetts Institute of Technology (MIT)

http://gdp.csail.mit.edu

Welcome!

The MIT **Geometric Data Processing Group** studies geometric problems in computer graphics, computer vision, machine learning, and other disciplines.

Our team includes students and researchers spanning a variety of disciplines, from theoretical mathematics to applications in engineering and software development. We enthusiastically welcome collaborators and staff at all levels and encourage interested parties to contact us with ideas, challenging problems, and avenues for joint research.

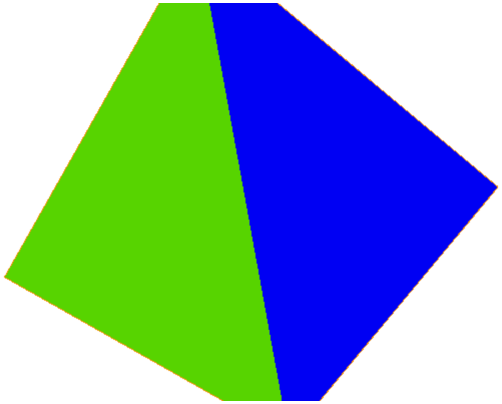
News

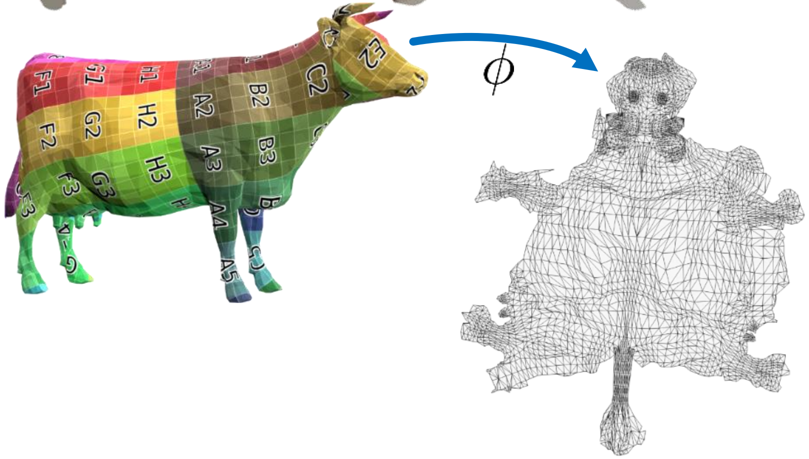
New website 10/21/18

Please contact Justin with comments or edits

Affiliations: EECS, CSAIL, Metric Geometry & Gerrymandering Group, Center for Computational Engineering, MIT-IBM Watson AI Lab, Toyota-CSAIL Joint Research Center

Shape Analysis: Typical Tasks

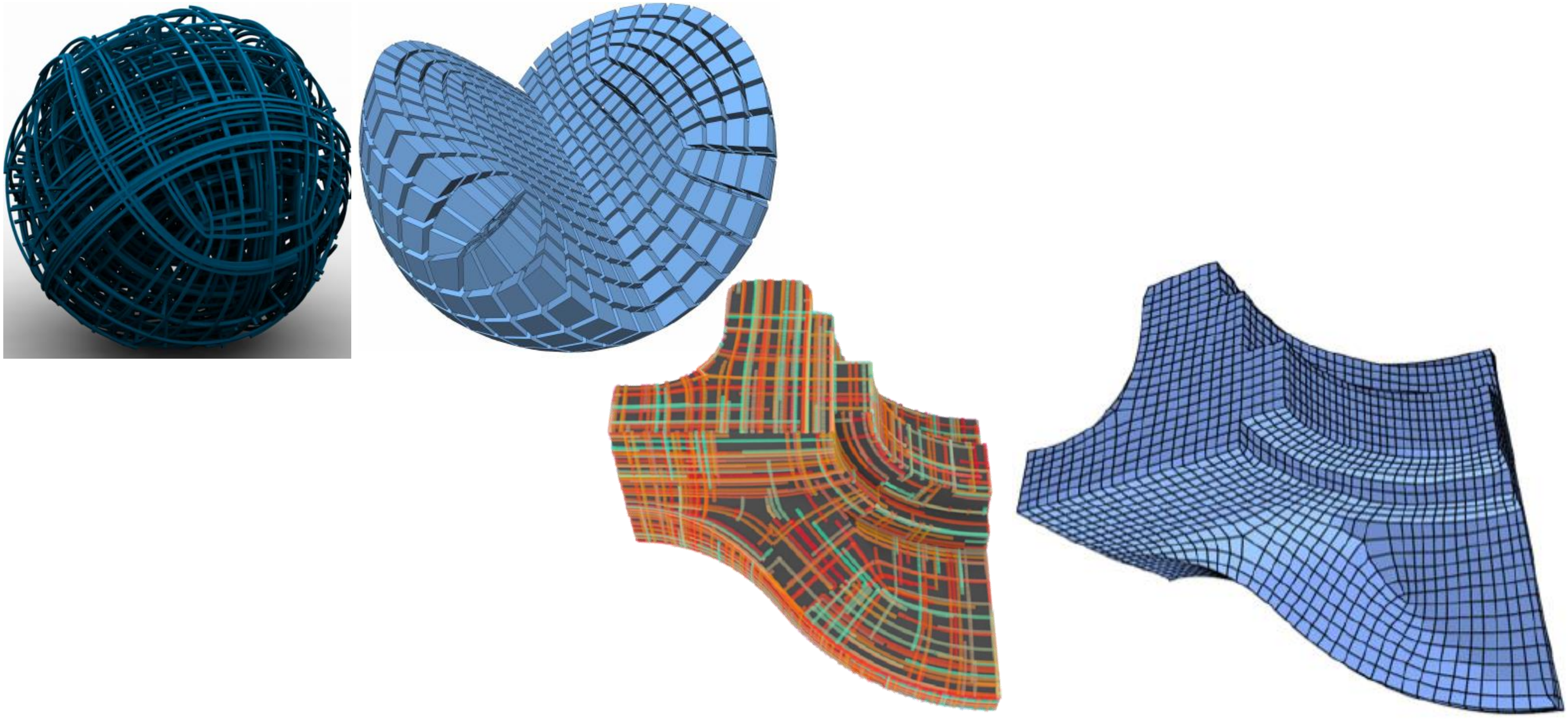




How do you embed domains into one another efficiently and with low distortion?

Claici et al. "Isometry-Aware Preconditioning for Mesh Parameterization." SGP 2017, London.
Li et al. "OptCuts: Joint Optimization of Surface Cuts and Parameterization." SIGGRAPH Asia 2018, Tokyo.
Gehre et al. "Interactive Curve Constrained Functional Maps." SGP 2018, Paris.

Shape Analysis: Typical Tasks



How can we tile a shape with simpler elements?

Solomon, Vaxman, and Bommes. "Boundary Element Octahedral Fields in Volumes." TOG 2018.
Zhang et al. "Spherical Harmonic Frames for Feature-Aligned Cross-Fields." Submitted.

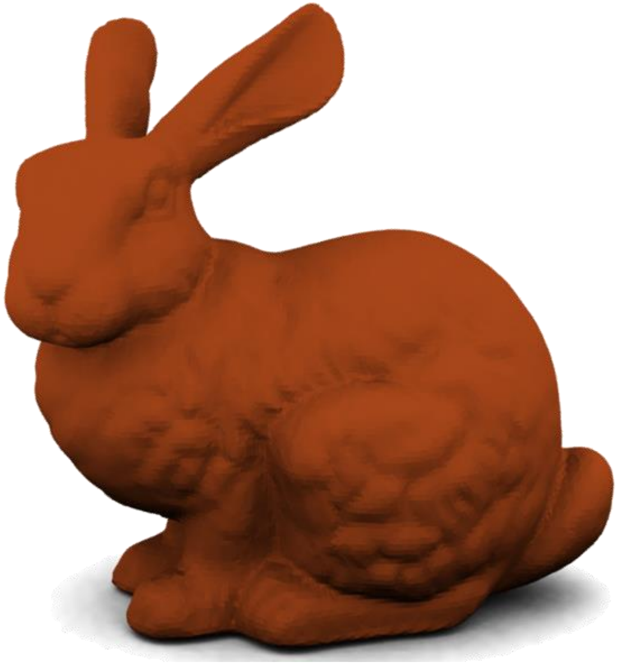
Shape Analysis: Typical Tasks

$$\boxed{I_\Omega^{\mathrm{TV}}(t) := \left\{ \begin{array}{rl} \min_{f \in L^1(\mathbb{R}^n)} & \mathrm{TV}[f] \\ \text{subject to} & \int_{\mathbb{R}^n} f(x)\,dx = t \\ & 0 \leq f \leq \mathbb{1}_\Omega \end{array} \right.}$$

How do we stabilize classical geometric measurements?

DeFord, Lavenant, Schutzman, and Solomon.
"Total Variation Isoperimetric Profiles." SIAM SIAGA, to appear.

Today’s Research Thread

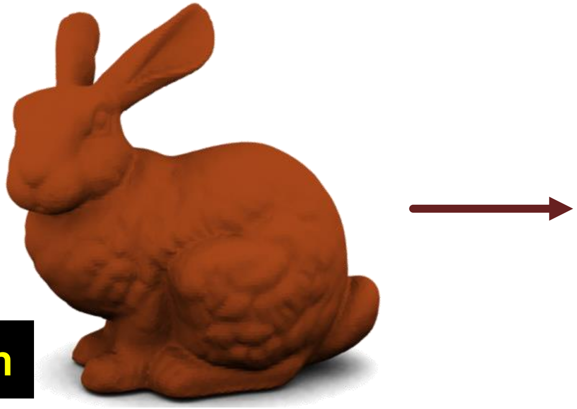


Meshes

Point clouds

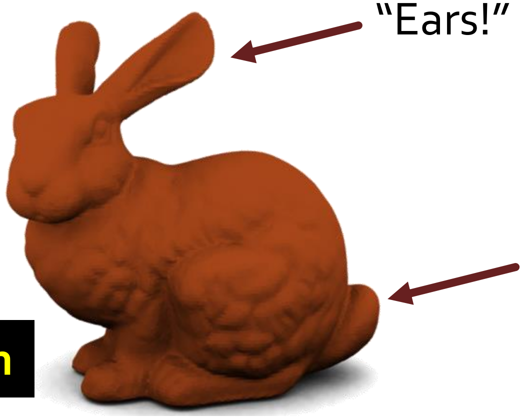
(Deep) learning on geometric data

Typical Tasks



Classification

“Bunny!”

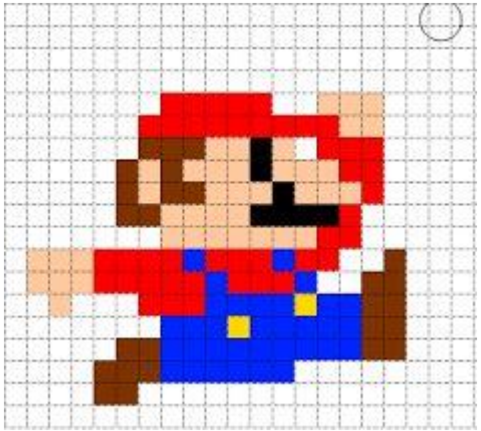


Segmentation

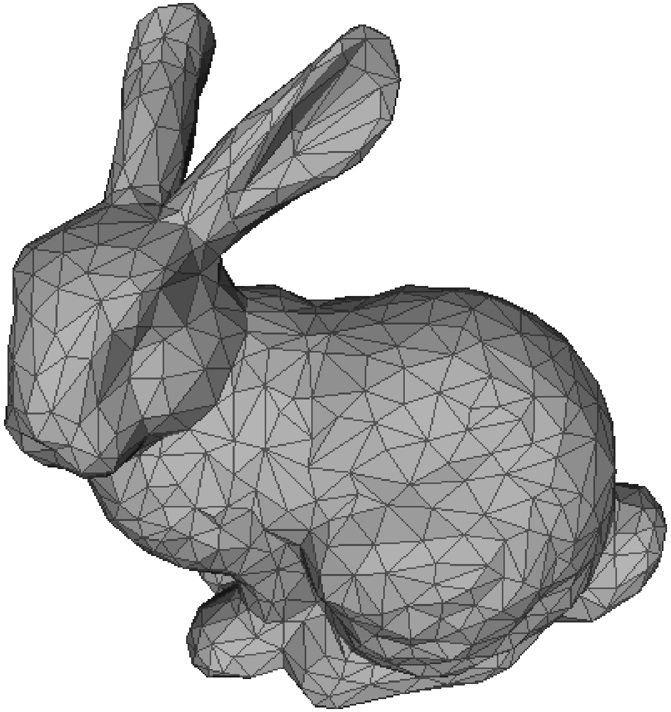
“Ears!”

“Tail!”

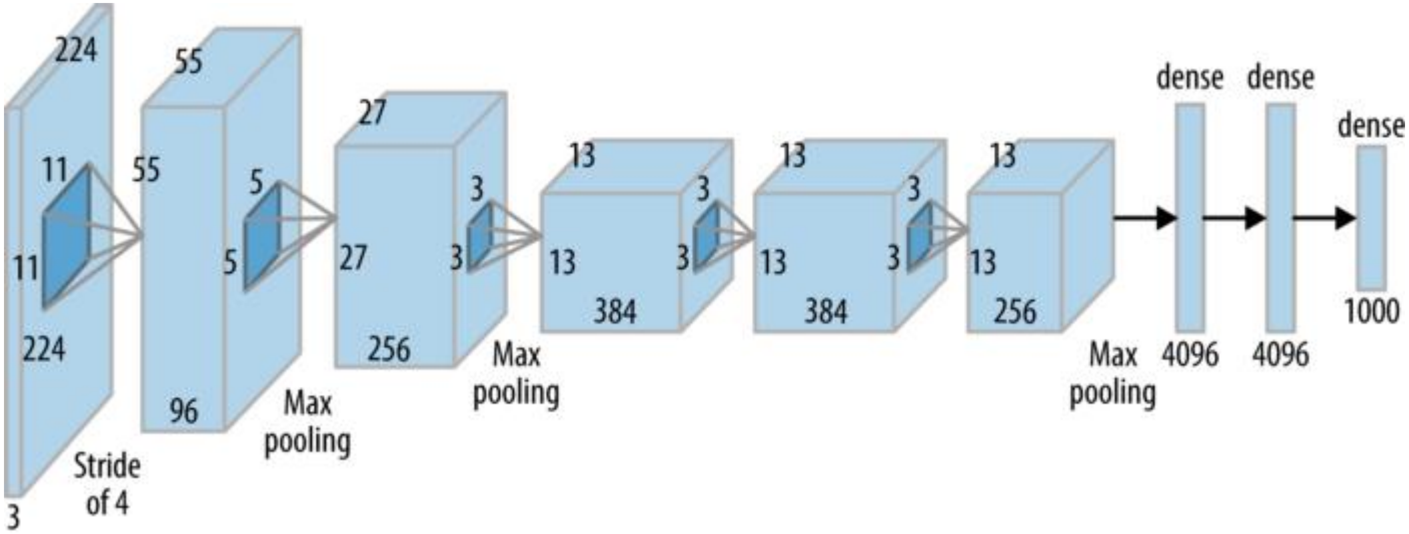
The Challenge



$\neq$



Typical Image-Based Learning

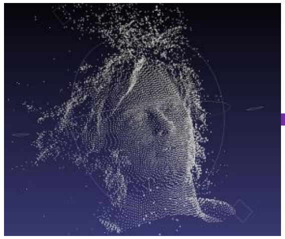


AlexNet

Krizhevsky, Sutskever, & Hinton:
"ImageNet classification with deep convolutional neural networks"

https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96

3D Learning: What We Want to Avoid



3D data

Shoehorn unit

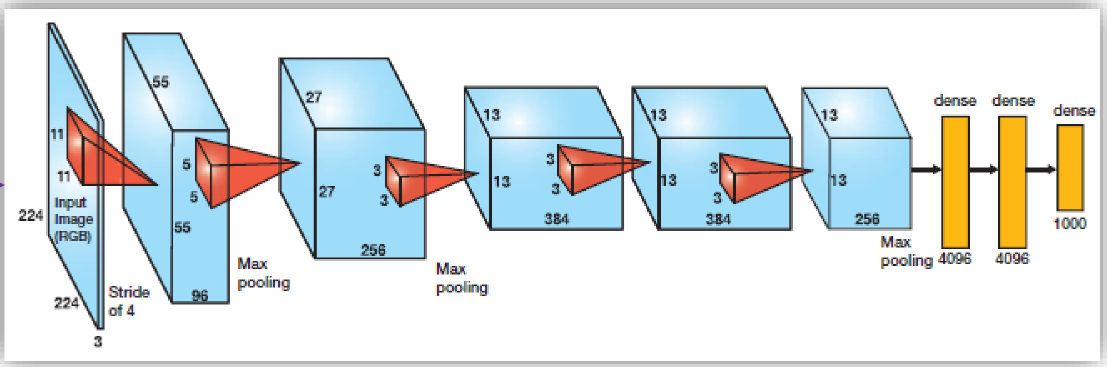
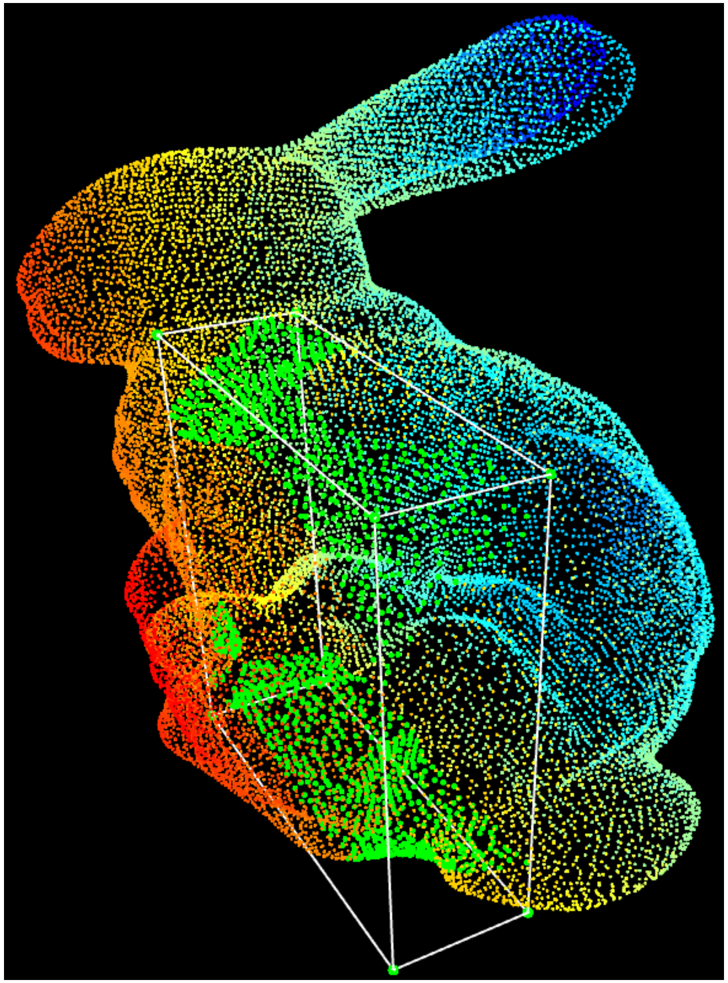


Image-based learning

Point Cloud Learning



Classification

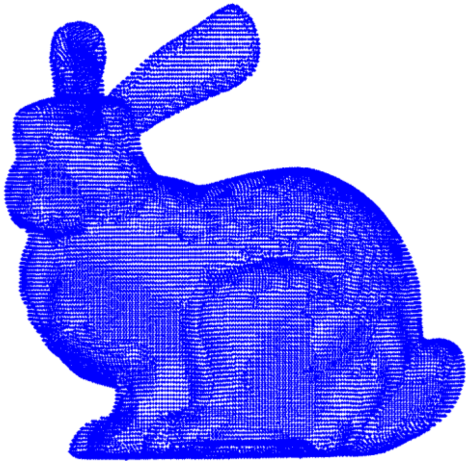
Map point cloud to a label in $\mathbb{R}^n$

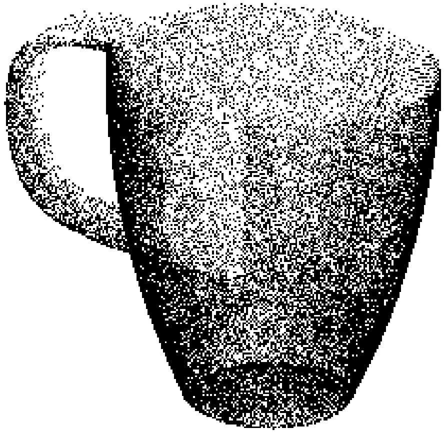
Segmentation

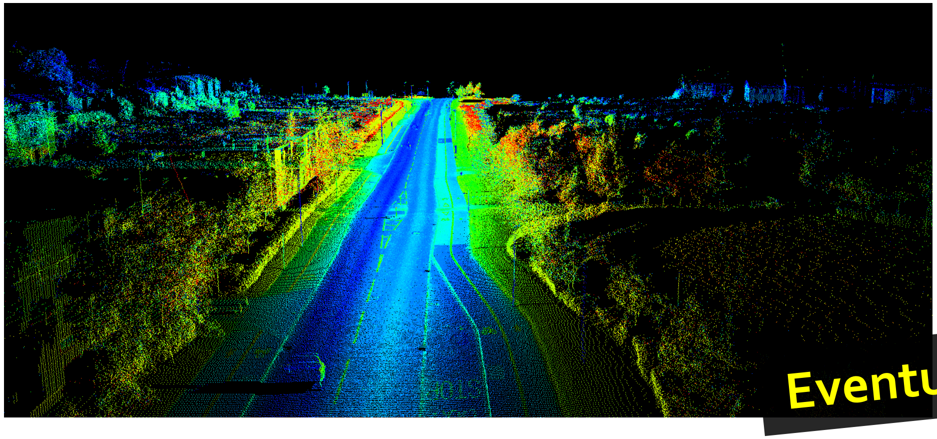
Map each point to a label in $\mathbb{R}^n$

http://paradise.caltech.edu/~yli/software/pceditor.html

Input



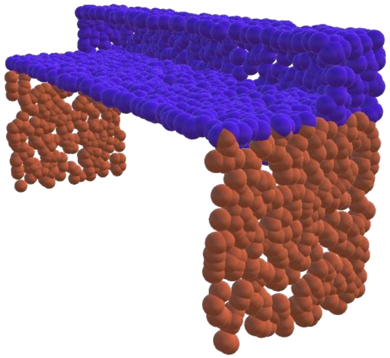


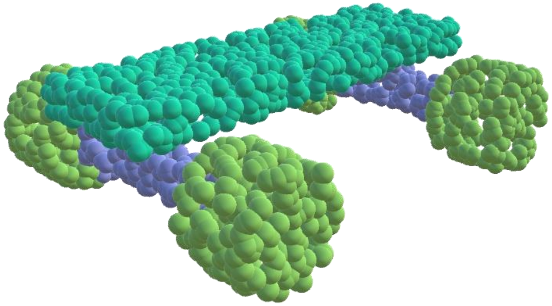


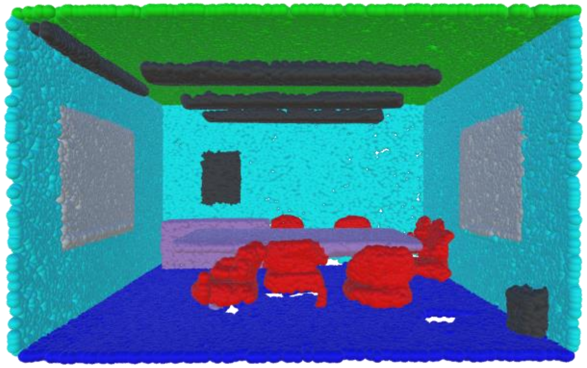
Eventual goal

Unordered set of points $\{(x_i, y_i, z_i)\}_{i=1}^{N}$

Challenges

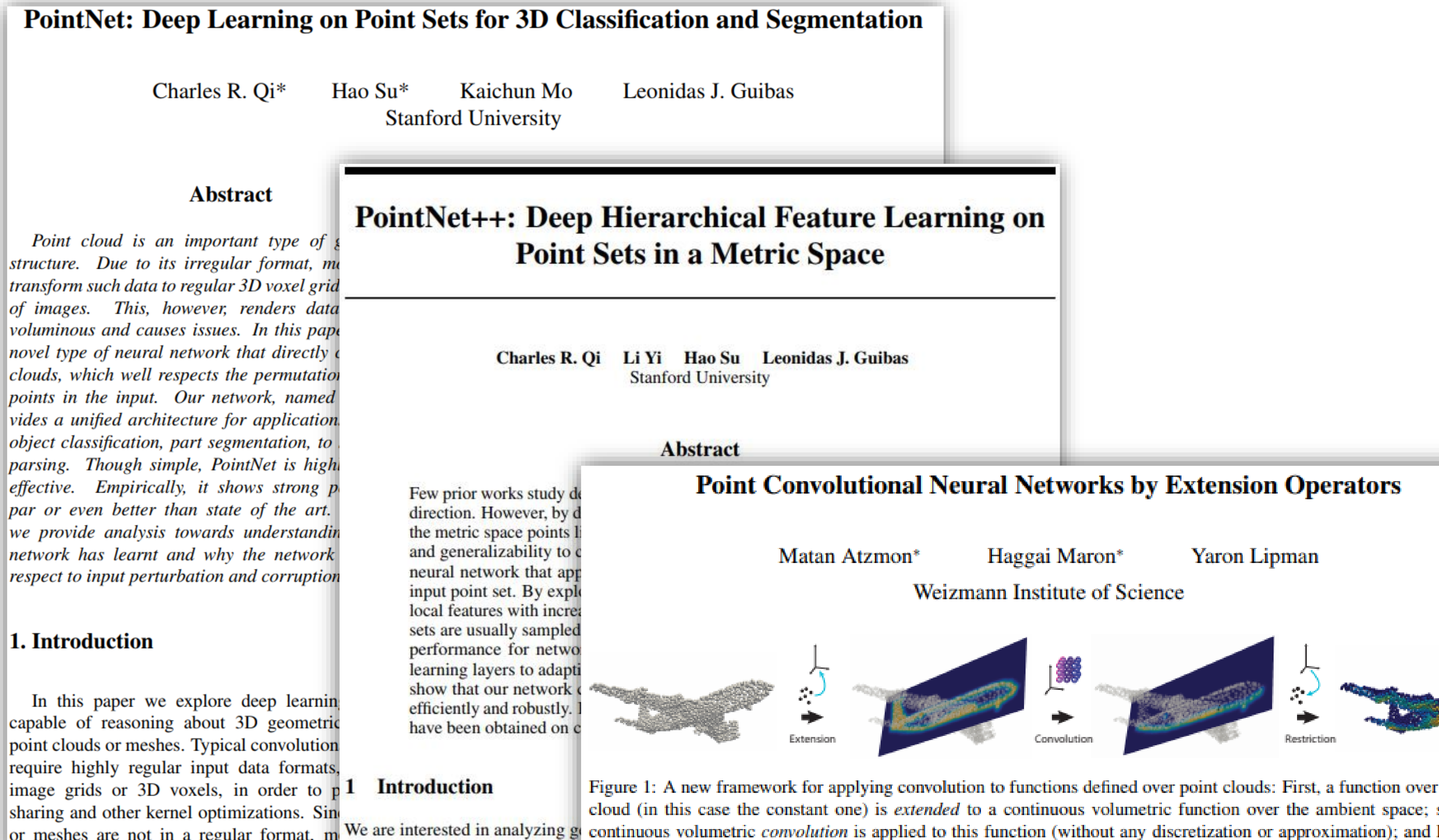






- Point clouds are unordered and unstructured
- Cannot parameterize patches
- No convolution
- Need a means for points to interact

Some Recent Options



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

Abstract

Point cloud is an important type of g
structure. Due to its irregular format, m
transform such data to regular 3D voxel grid
of images. This, however, renders data
voluminous and causes issues. In this pape
novel type of neural network that directly c
clouds, which well respects the permutatio
points in the input. Our network, named
vides a unified architecture for application
object classification, part segmentation, to
parsing. Though simple, PointNet is highl
effective. Empirically, it shows strong p
par or even better than state of the art.
we provide analysis towards understandin
network has learnt and why the network
respect to input perturbation and corruption

1. Introduction

In this paper we explore deep learnin
capable of reasoning about 3D geometric
point clouds or meshes. Typical convolution
require highly regular input data formats,
image grids or 3D voxels, in order to p
sharing and other kernel optimizations. Sin
or meshes are not in a regular format, m

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Charles R. Qi Li Yi Hao Su Leonidas J. Guibas
Stanford University

Abstract

Few prior works study d
direction. However, by d
the metric space points l
and generalizability to c
neural network that app
input point set. By expl
local features with incre
sets are usually sampled
performance for netwo
learning layers to adapti
show that our network c
efficiently and robustly. I
have been obtained on c

1 Introduction

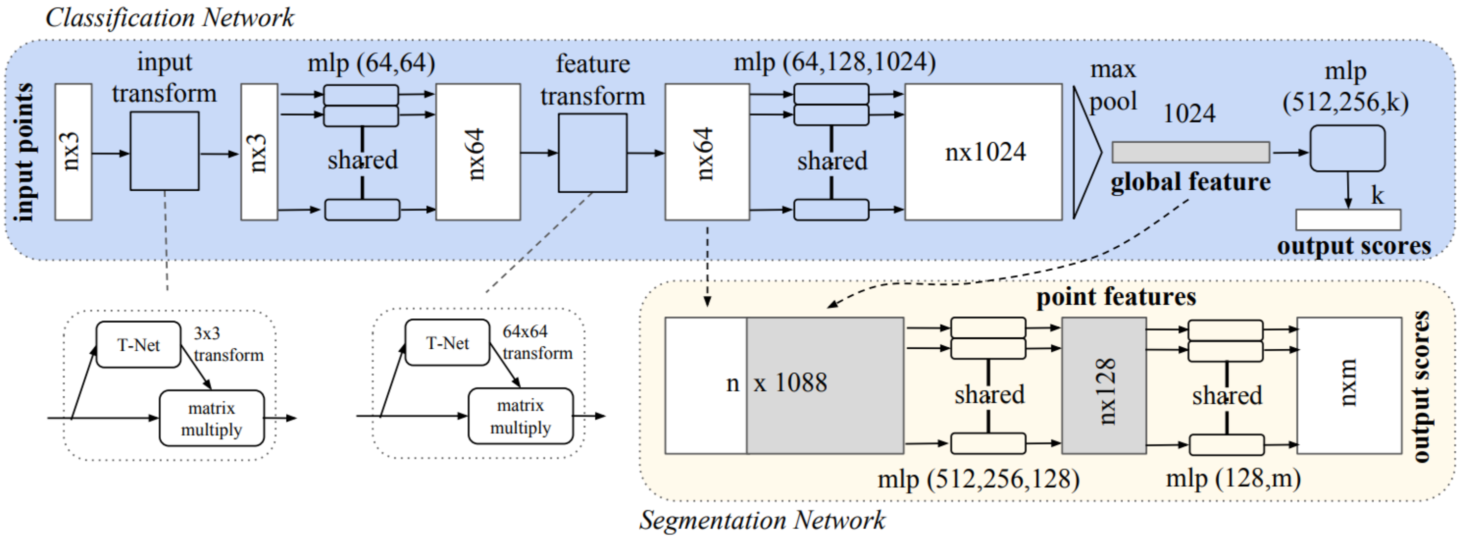
We are interested in analyzing g

Point Convolutional Neural Networks by Extension Operators

Matan Atzmon* Haggai Maron* Yaron Lipman
Weizmann Institute of Science

Figure 1: A new framework for applying convolution to functions defined over point clouds: First, a function over
cloud (in this case the constant one) is *extended* to a continuous volumetric function over the ambient space;
continuous volumetric *convolution* is applied to this function (without any discretization or approximation); and

Remarkable First Step



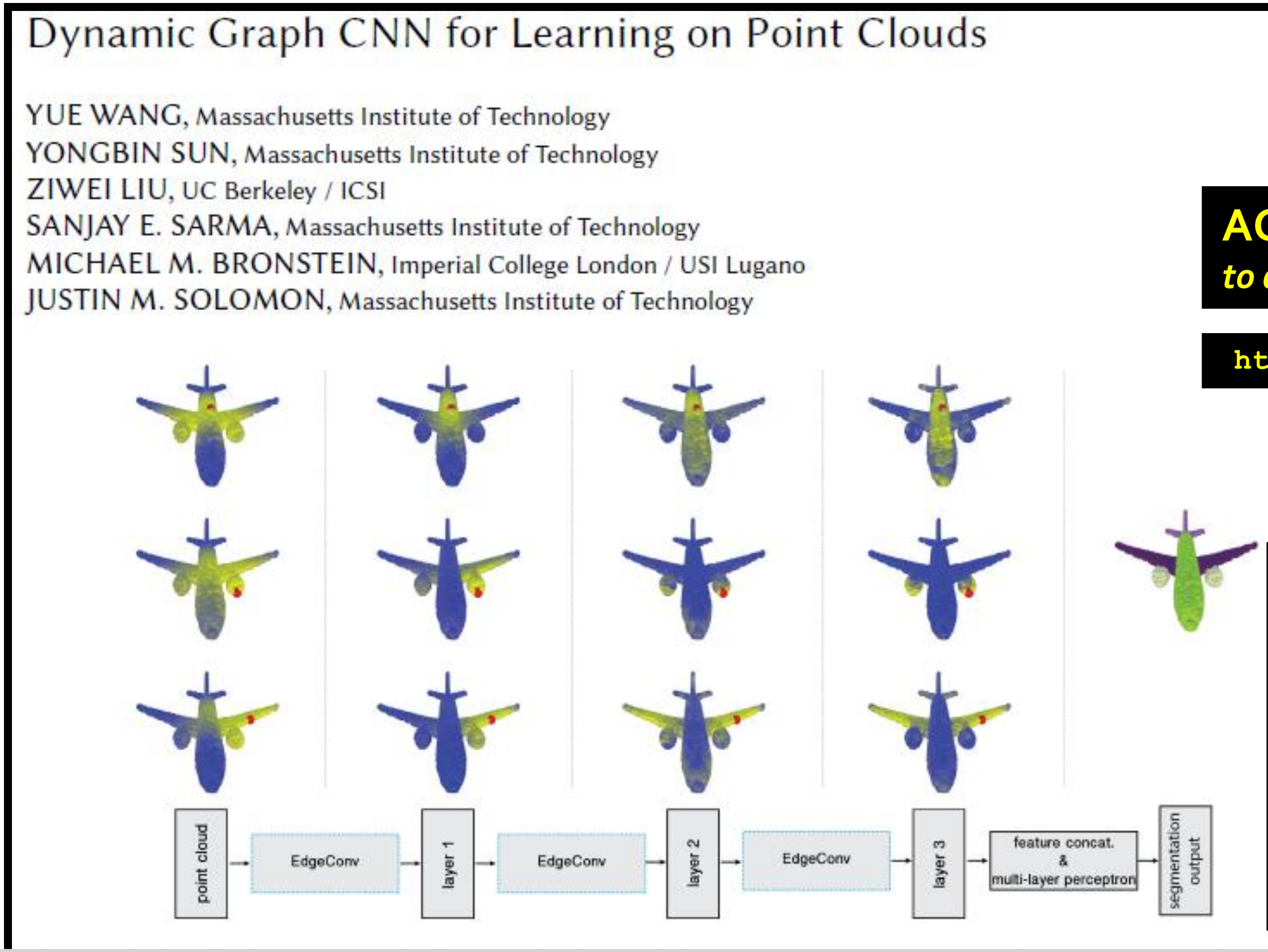
Embeds points individually

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Qi, Su, Mo, & Guibas; CVPR 2017

PointNet architecture

DGCNN

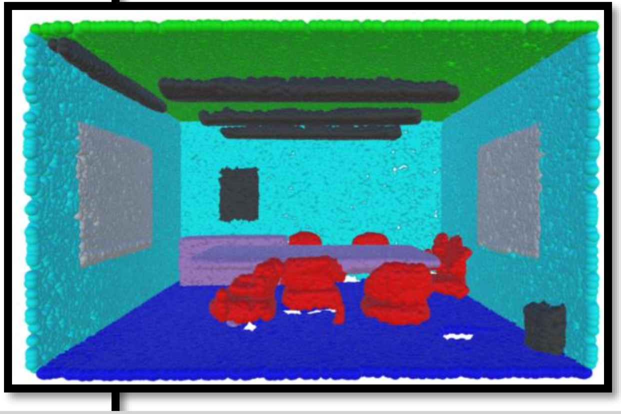


Dynamic Graph CNN for Learning on Point Clouds

YUE WANG, Massachusetts Institute of Technology
YONGBIN SUN, Massachusetts Institute of Technology
ZIWEI LIU, UC Berkeley / ICSI
SANJAY E. SARMA, Massachusetts Institute of Technology
MICHAEL M. BRONSTEIN, Imperial College London / USI Lugano
JUSTIN M. SOLOMON, Massachusetts Institute of Technology

ACM Transactions on Graphics
to appear

`https://github.com/WangYueFt/dgcnn`



Learning on point clouds

Desiderata

- Order invariance

  No natural order for list of points

- Captures global information

  Combine information over entire shape

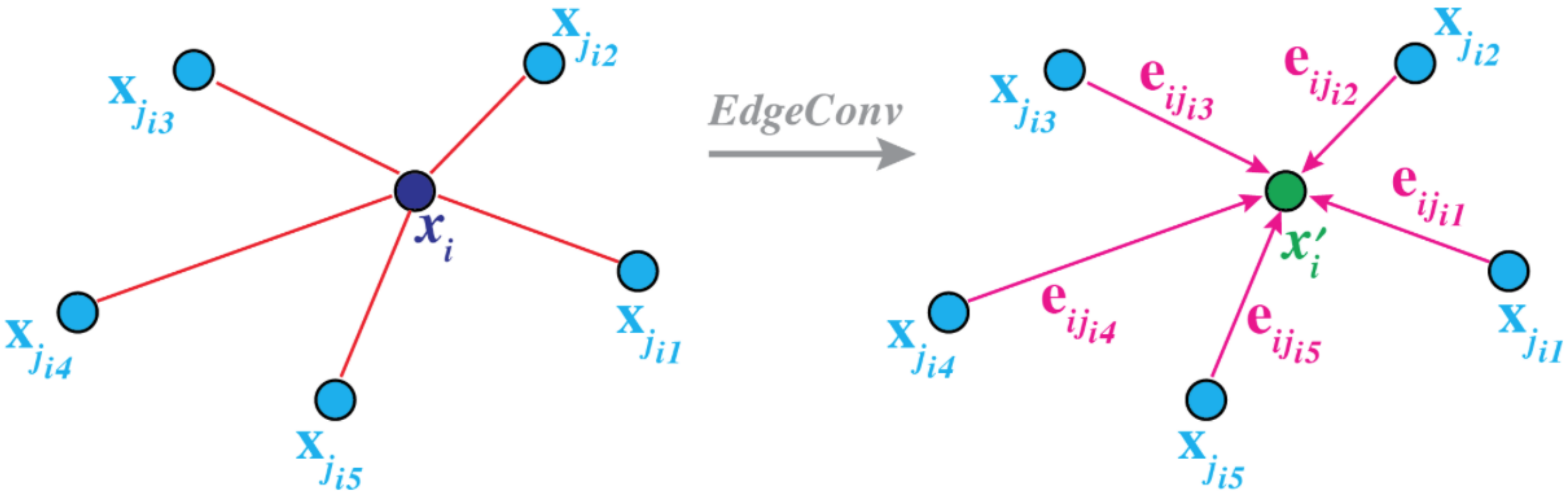
- Leverages local neighborhoods

  Curvature, local features relevant

- Large receptive field

  Some version of density independence

Introducing: EdgeConv



- **Feature per edge**

  Concatenate two endpoints

- **Feed-forward NN to transform**

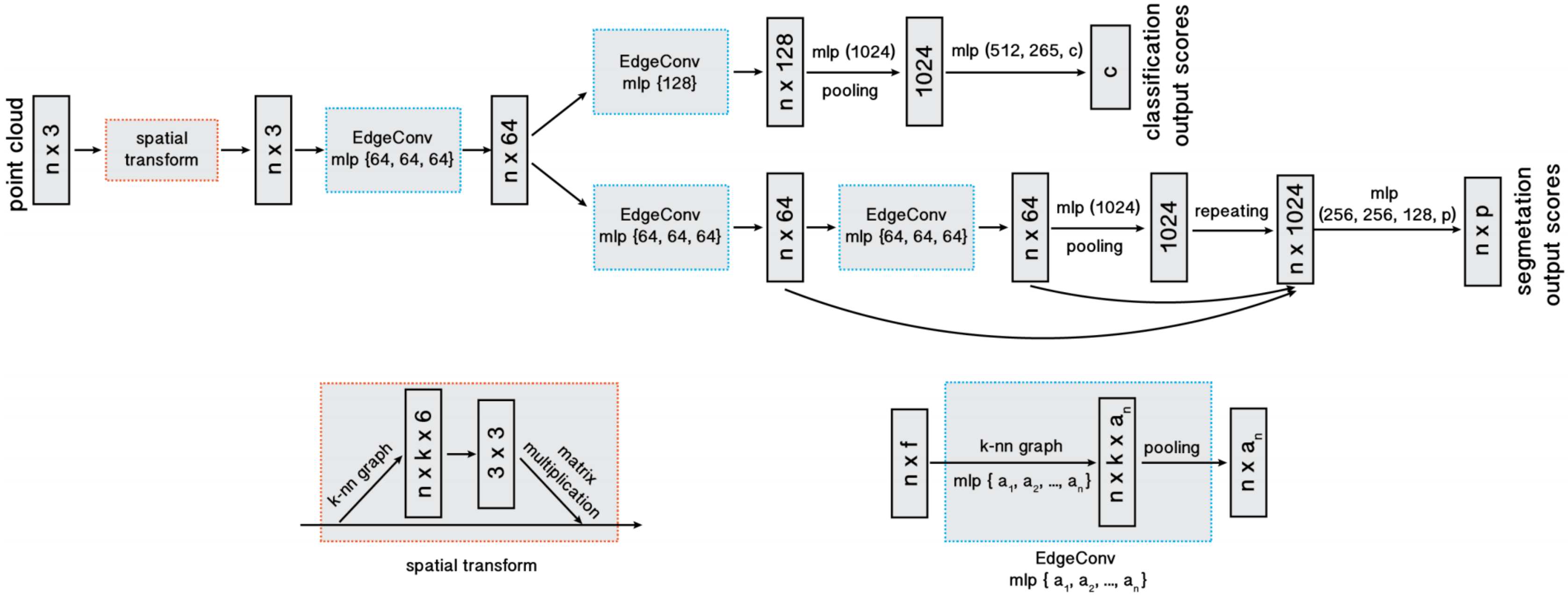
  High-dimensional point per edge

- **Symmetric aggregation**

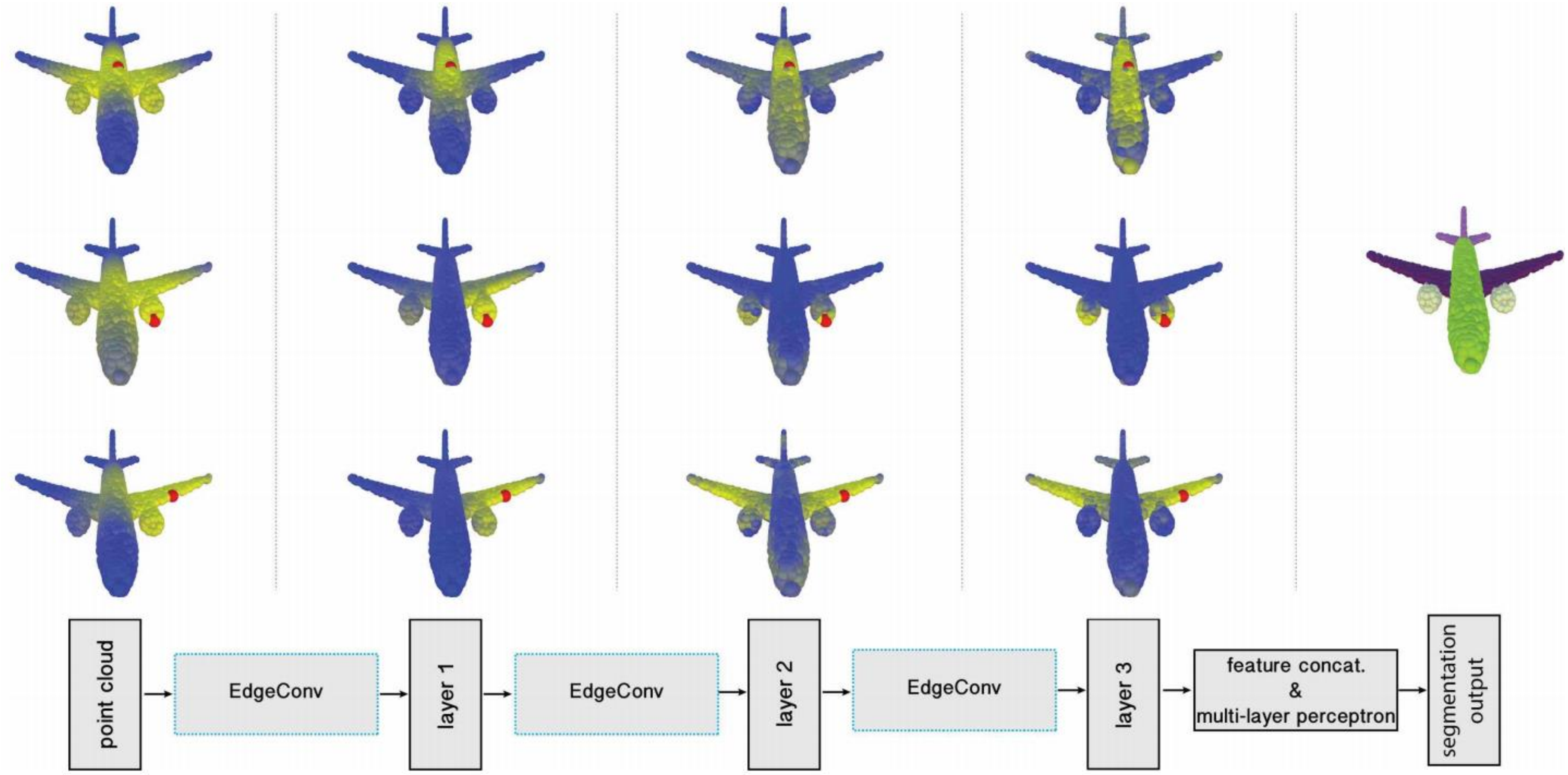
  Back to center point

Dynamic Graph

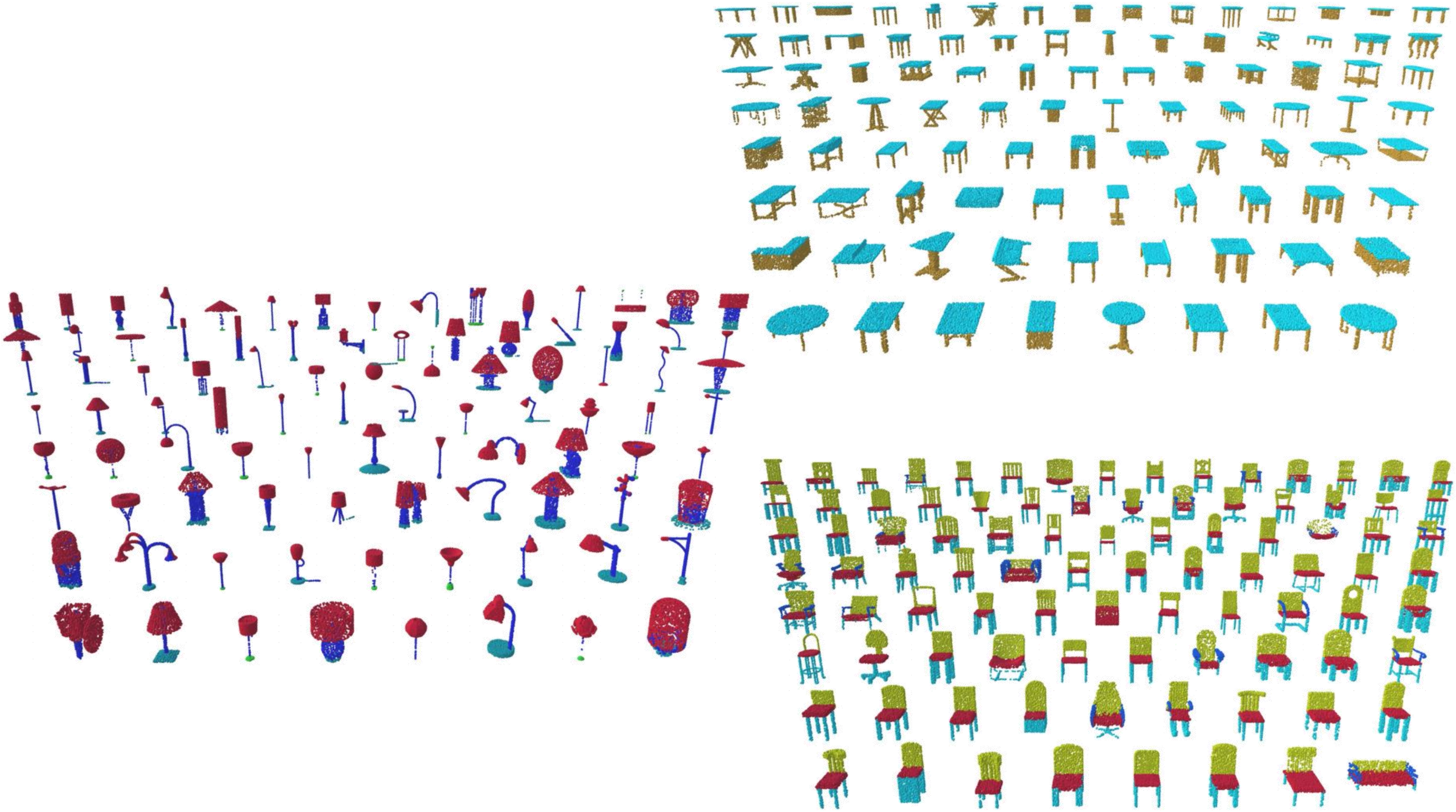
- Stack EdgeConv layers
- Recompute KNN before each layer
- Notion of “nearby” changes each layer



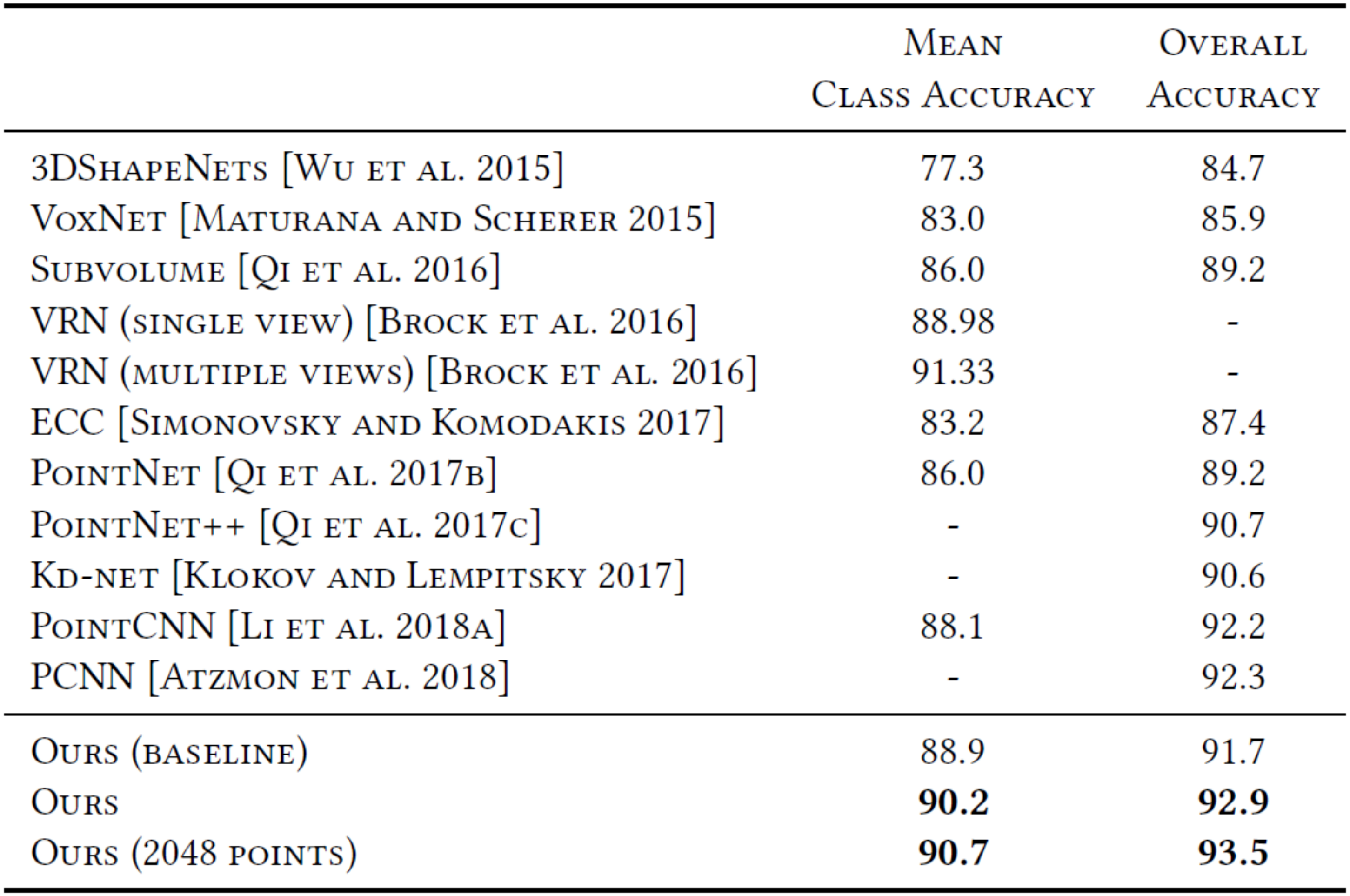
Full Segmentation Pipeline



Segmentation



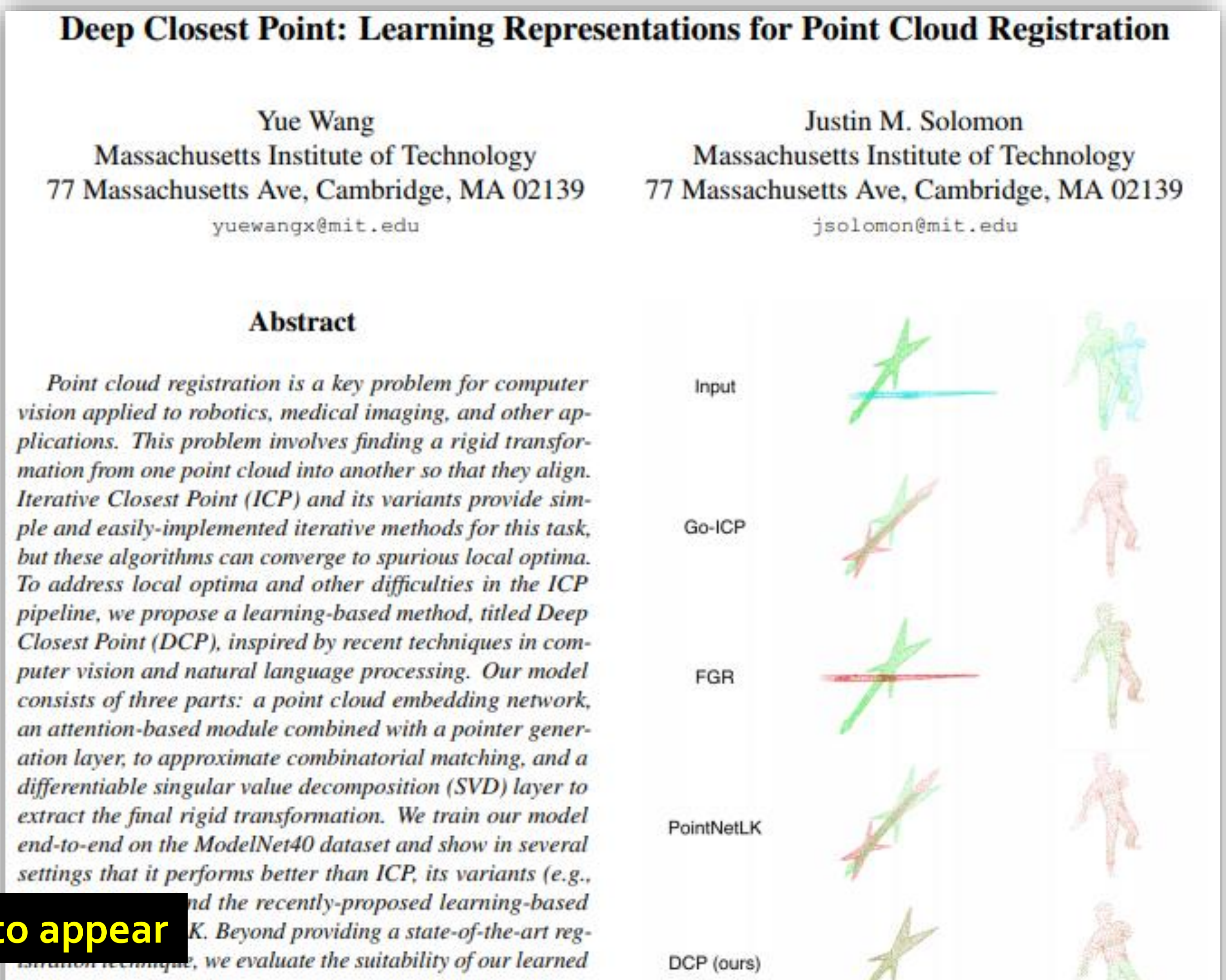
Evaluation



	Mean Class Accuracy	Overall Accuracy
3DShapeNets [Wu et al. 2015]	77.3	84.7
VoxNet [Maturana and Scherer 2015]	83.0	85.9
Subvolume [Qi et al. 2016]	86.0	89.2
VRN (single view) [Brock et al. 2016]	88.98	-
VRN (multiple views) [Brock et al. 2016]	91.33	-
ECC [Simonovsky and Komodakis 2017]	83.2	87.4
PointNet [Qi et al. 2017b]	86.0	89.2
PointNet++ [Qi et al. 2017c]	-	90.7
Kd-net [Klokov and Lempitsky 2017]	-	90.6
PointCNN [Li et al. 2018a]	88.1	92.2
PCNN [Atzmon et al. 2018]	-	92.3
Ours (baseline)	88.9	91.7
Ours	**90.2**	**92.9**
Ours (2048 points)	**90.7**	**93.5**

Table 2. Classification results on ModelNet40.

Recent Extension



Deep Closest Point: Learning Representations for Point Cloud Registration

Yue Wang
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
yuewangx@mit.edu

Justin M. Solomon
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
jsolomon@mit.edu

Abstract

Point cloud registration is a key problem for computer vision applied to robotics, medical imaging, and other applications. This problem involves finding a rigid transformation from one point cloud into another so that they align. Iterative Closest Point (ICP) and its variants provide simple and easily-implemented iterative methods for this task, but these algorithms can converge to spurious local optima. To address local optima and other difficulties in the ICP pipeline, we propose a learning-based method, titled Deep Closest Point (DCP), inspired by recent techniques in computer vision and natural language processing. Our model consists of three parts: a point cloud embedding network, an attention-based module combined with a pointer generation layer, to approximate combinatorial matching, and a differentiable singular value decomposition (SVD) layer to extract the final rigid transformation. We train our model end-to-end on the ModelNet40 dataset and show in several settings that it performs better than ICP, its variants (e.g., nd the recently-proposed learning-based K. Beyond providing a state-of-the-art registration technique, we evaluate the suitability of our learned

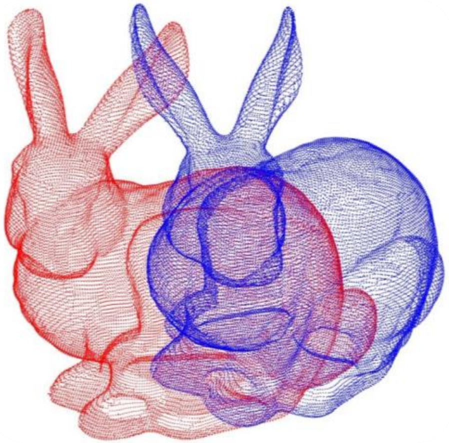
ICCV 2019, to appear

Classical Iterative Closest Point

- **Choose** e.g. 1000 random points
- **Match** each to closest point on other scan
- **Reject** pairs with distance > *k* times median
- **Minimize**

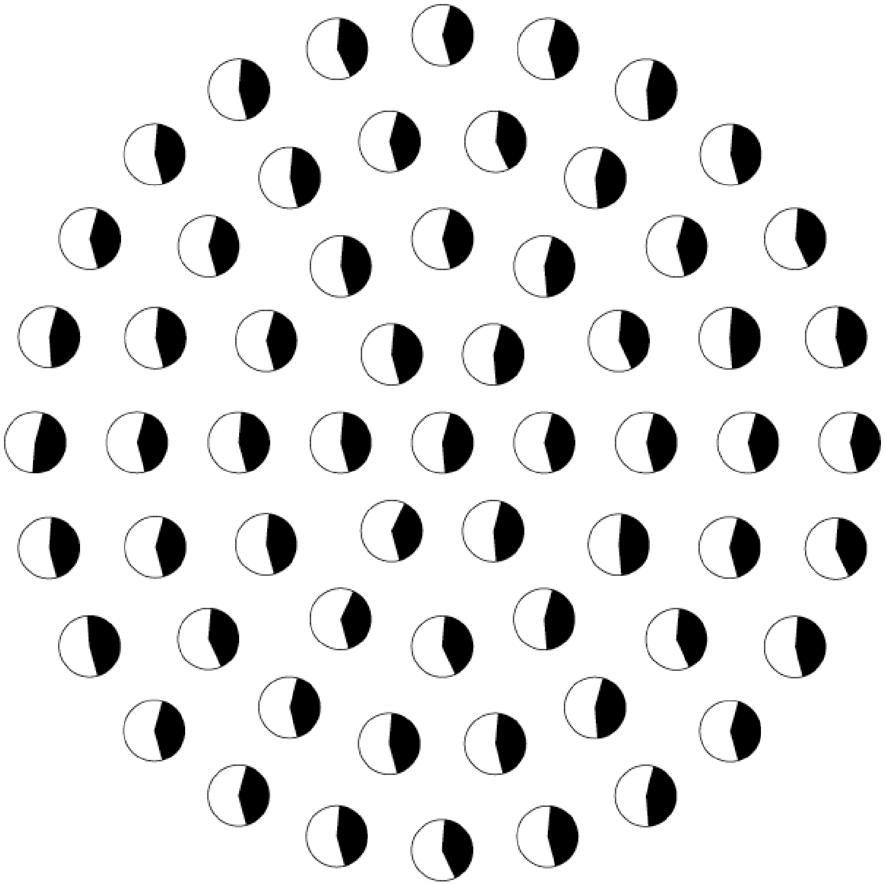
$$E[R,t] := \sum_i \|Rp_i + t - q_i\|^2$$

- **Iterate**



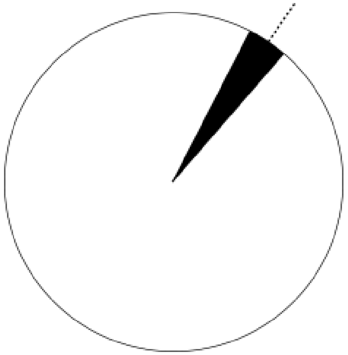
"A method for registration of 3-D shapes."
Besl and McKay, PAMI 1992.

Local Optima



Translation in xz plane

Rotation about y

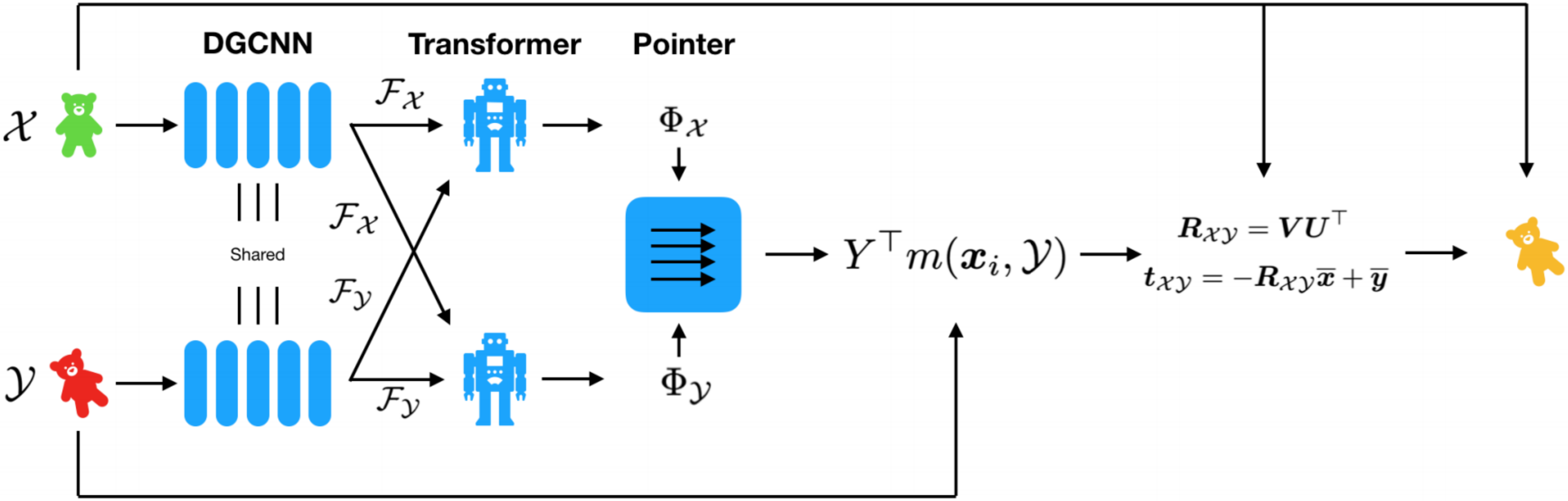


■ Converges

□ Does not converge

Slide courtesy N. Mitra

Our Approach


DGCNN
Transformer
Pointer
$\mathcal{X}$
$\mathcal{F}_\mathcal{X}$
$\Phi_\mathcal{X}$
$\mathcal{F}_\mathcal{X}$
Shared
$\mathcal{F}_\mathcal{Y}$
$Y^\top m(\boldsymbol{x}_i, \mathcal{Y})$
$\boldsymbol{R}_{\mathcal{XY}} = \boldsymbol{V}\boldsymbol{U}^\top$
$\boldsymbol{t}_{\mathcal{XY}} = -\boldsymbol{R}_{\mathcal{XY}}\overline{\boldsymbol{x}} + \overline{\boldsymbol{y}}$
$\mathcal{Y}$
$\mathcal{F}_\mathcal{Y}$
$\Phi_\mathcal{Y}$

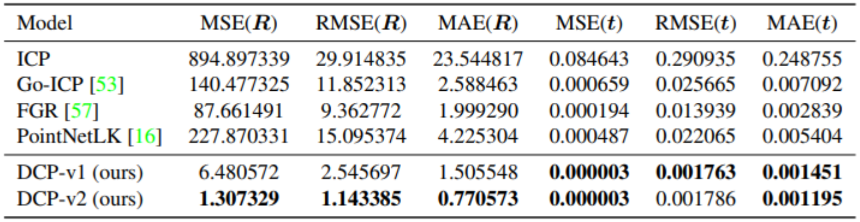
Choose closest points in feature space

Training data: Synthetically rotated shape pairs

Features are self-supervised!

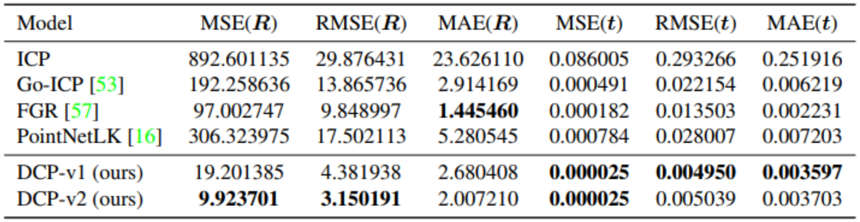
DGCNN learns features

Results



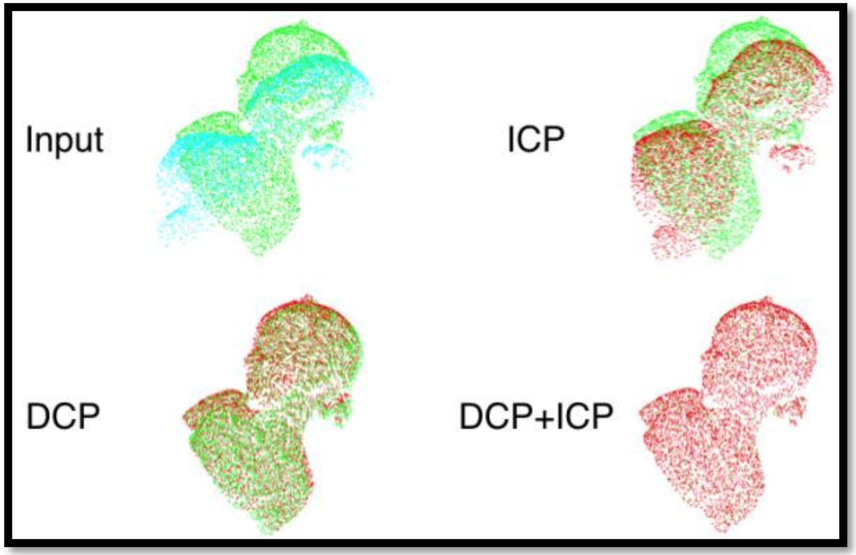
Model	MSE(*R*)	RMSE(*R*)	MAE(*R*)	MSE(*t*)	RMSE(*t*)	MAE(*t*)
ICP	894.897339	29.914835	23.544817	0.084643	0.290935	0.248755
Go-ICP [53]	140.477325	11.852313	2.588463	0.000659	0.025665	0.007092
FGR [57]	87.661491	9.362772	1.999290	0.000194	0.013939	0.002839
PointNetLK [16]	227.870331	15.095374	4.225304	0.000487	0.022065	0.005404
DCP-v1 (ours)	6.480572	2.545697	1.505548	**0.000003**	**0.001763**	**0.001451**
DCP-v2 (ours)	**1.307329**	**1.143385**	**0.770573**	**0.000003**	0.001786	**0.001195**

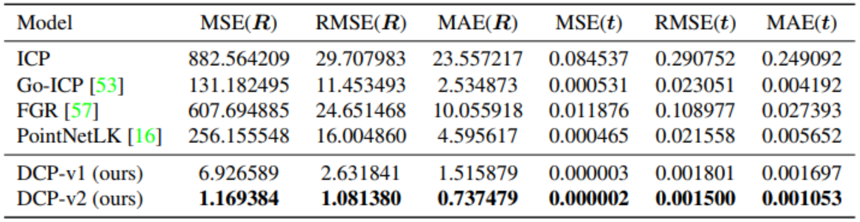
Table 1. ModelNet40: Test on unseen point clouds



Model	MSE(*R*)	RMSE(*R*)	MAE(*R*)	MSE(*t*)	RMSE(*t*)	MAE(*t*)
ICP	892.601135	29.876431	23.626110	0.086005	0.293266	0.251916
Go-ICP [53]	192.258636	13.865736	2.914169	0.000491	0.022154	0.006219
FGR [57]	97.002747	9.848997	**1.445460**	0.000182	0.013503	0.002231
PointNetLK [16]	306.323975	17.502113	5.280545	0.000784	0.028007	0.007203
DCP-v1 (ours)	19.201385	4.381938	2.680408	**0.000025**	**0.004950**	**0.003597**
DCP-v2 (ours)	**9.923701**	**3.150191**	2.007210	**0.000025**	0.005039	0.003703

Table 2. ModelNet40: Test on unseen categories

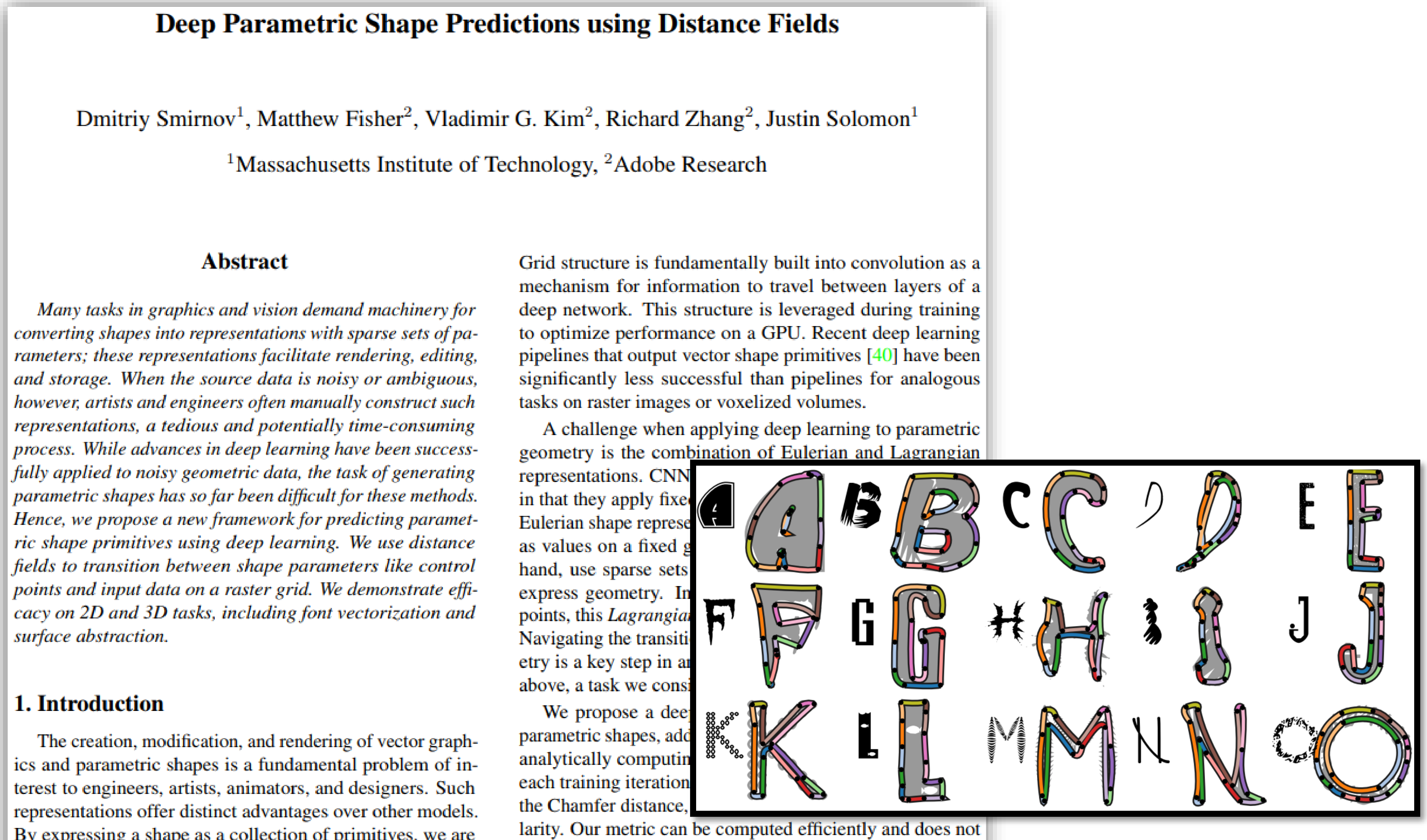




Model	MSE(*R*)	RMSE(*R*)	MAE(*R*)	MSE(*t*)	RMSE(*t*)	MAE(*t*)
ICP	882.564209	29.707983	23.557217	0.084537	0.290752	0.249092
Go-ICP [53]	131.182495	11.453493	2.534873	0.000531	0.023051	0.004192
FGR [57]	607.694885	24.651468	10.055918	0.011876	0.108977	0.027393
PointNetLK [16]	256.155548	16.004860	4.595617	0.000465	0.021558	0.005652
DCP-v1 (ours)	6.926589	2.631841	1.515879	0.000003	0.001801	0.001697
DCP-v2 (ours)	**1.169384**	**1.081380**	**0.737479**	**0.000002**	**0.001500**	**0.001053**

Table 3. ModelNet40: Test on objects with Gaussian noise

Different Task



Deep Parametric Shape Predictions using Distance Fields

Dmitriy Smirnov[1], Matthew Fisher[2], Vladimir G. Kim[2], Richard Zhang[2], Justin Solomon[1]

[1]Massachusetts Institute of Technology, [2]Adobe Research

Abstract

Many tasks in graphics and vision demand machinery for converting shapes into representations with sparse sets of parameters; these representations facilitate rendering, editing, and storage. When the source data is noisy or ambiguous, however, artists and engineers often manually construct such representations, a tedious and potentially time-consuming process. While advances in deep learning have been successfully applied to noisy geometric data, the task of generating parametric shapes has so far been difficult for these methods. Hence, we propose a new framework for predicting parametric shape primitives using deep learning. We use distance fields to transition between shape parameters like control points and input data on a raster grid. We demonstrate efficacy on 2D and 3D tasks, including font vectorization and surface abstraction.

1. Introduction

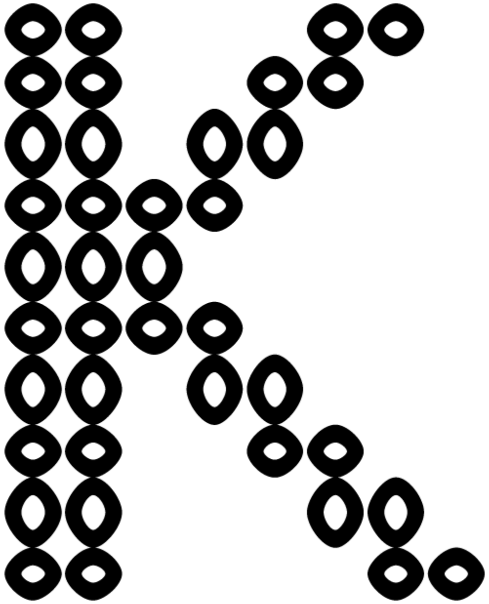
The creation, modification, and rendering of vector graphics and parametric shapes is a fundamental problem of interest to engineers, artists, animators, and designers. Such representations offer distinct advantages over other models. By expressing a shape as a collection of primitives, we are

Grid structure is fundamentally built into convolution as a mechanism for information to travel between layers of a deep network. This structure is leveraged during training to optimize performance on a GPU. Recent deep learning pipelines that output vector shape primitives [40] have been significantly less successful than pipelines for analogous tasks on raster images or voxelized volumes.

A challenge when applying deep learning to parametric geometry is the combination of Eulerian and Lagrangian representations. CNN
in that they apply fixe
Eulerian shape represe
as values on a fixed g
hand, use sparse sets
express geometry. In
points, this *Lagrangia*
Navigating the transiti
etry is a key step in a
above, a task we cons

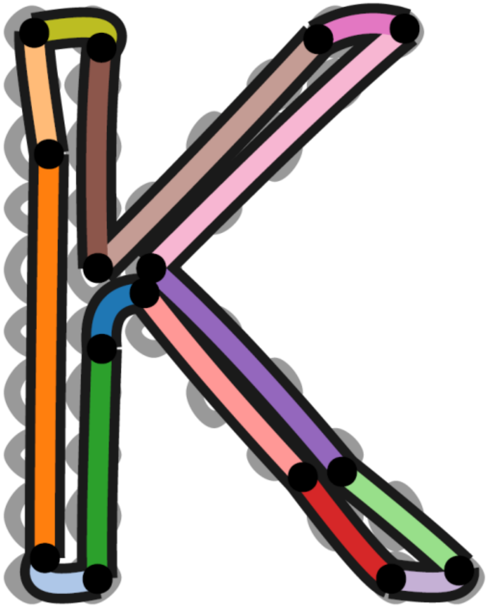
We propose a dee
parametric shapes, add
analytically computin
each training iteration
the Chamfer distance,
larity. Our metric can be computed efficiently and does not

High-Level Theme



Input data:

Eulerian representation



Input data:

Lagrangian representation

A Graceful Transition

$$\mathcal{L}_\Psi(d_A(\cdot), d_B(\cdot)) = \int_S \Psi[d_A(x), d_B(x)]\, dV(x)$$

Closed-form distances for Bézier curves, implicit primitives, and Boolean representations.

Generalizes:

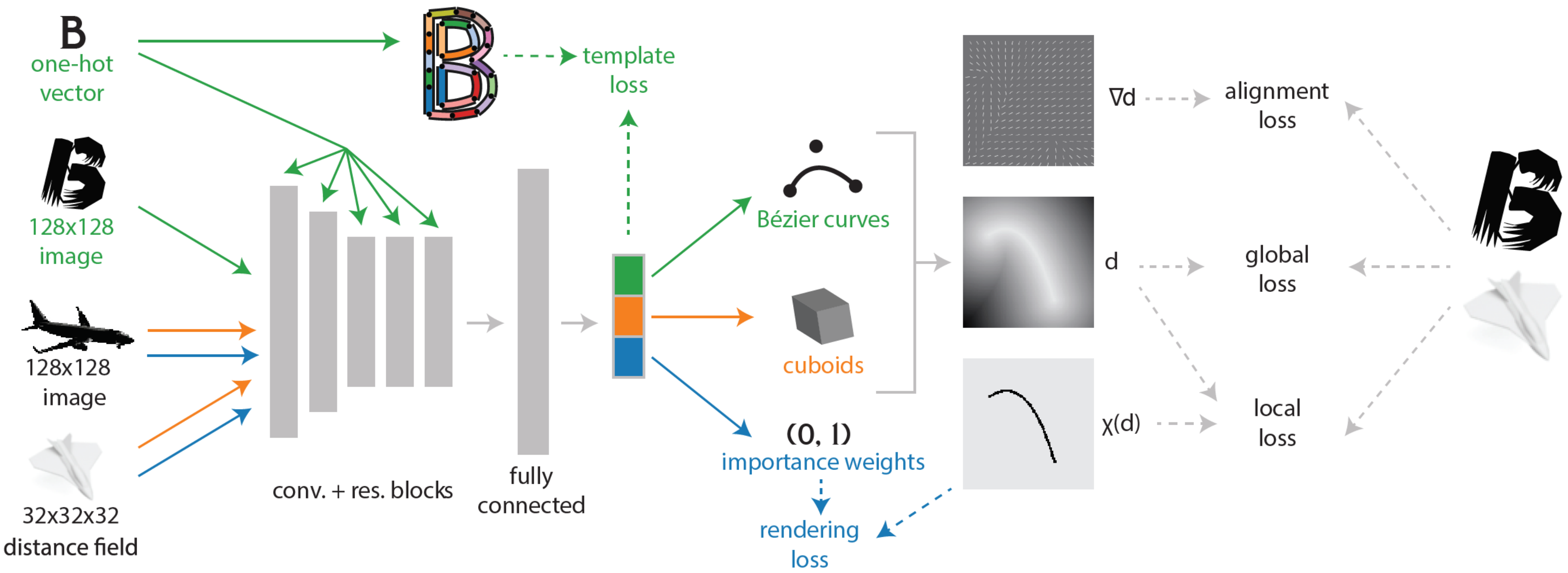
- Chamfer loss
- Global alignment
- Normal alignment

Bonus:

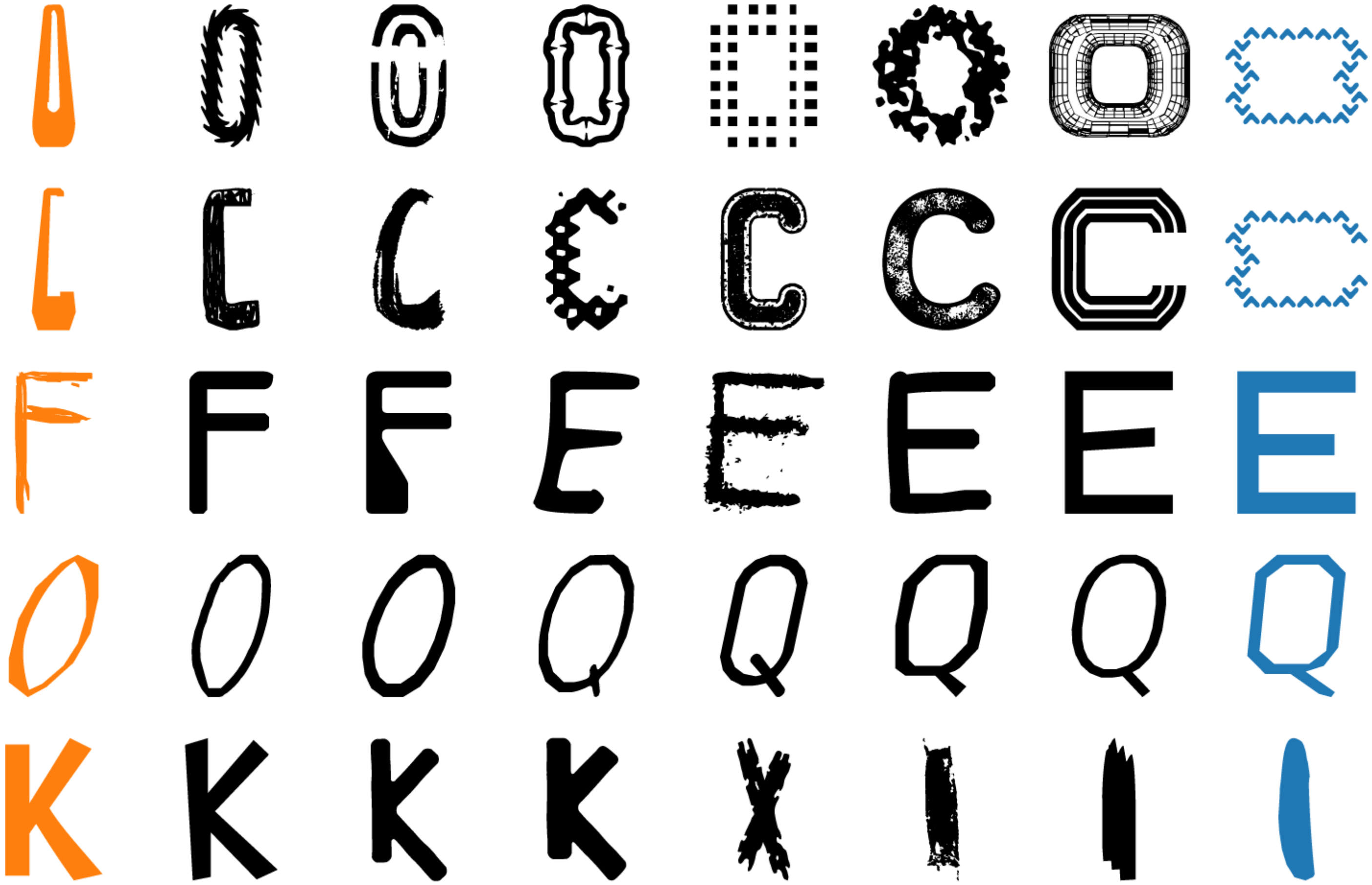
More efficient!

Alignment objectives are easy to evaluate from a distance field.

Learning Pipeline

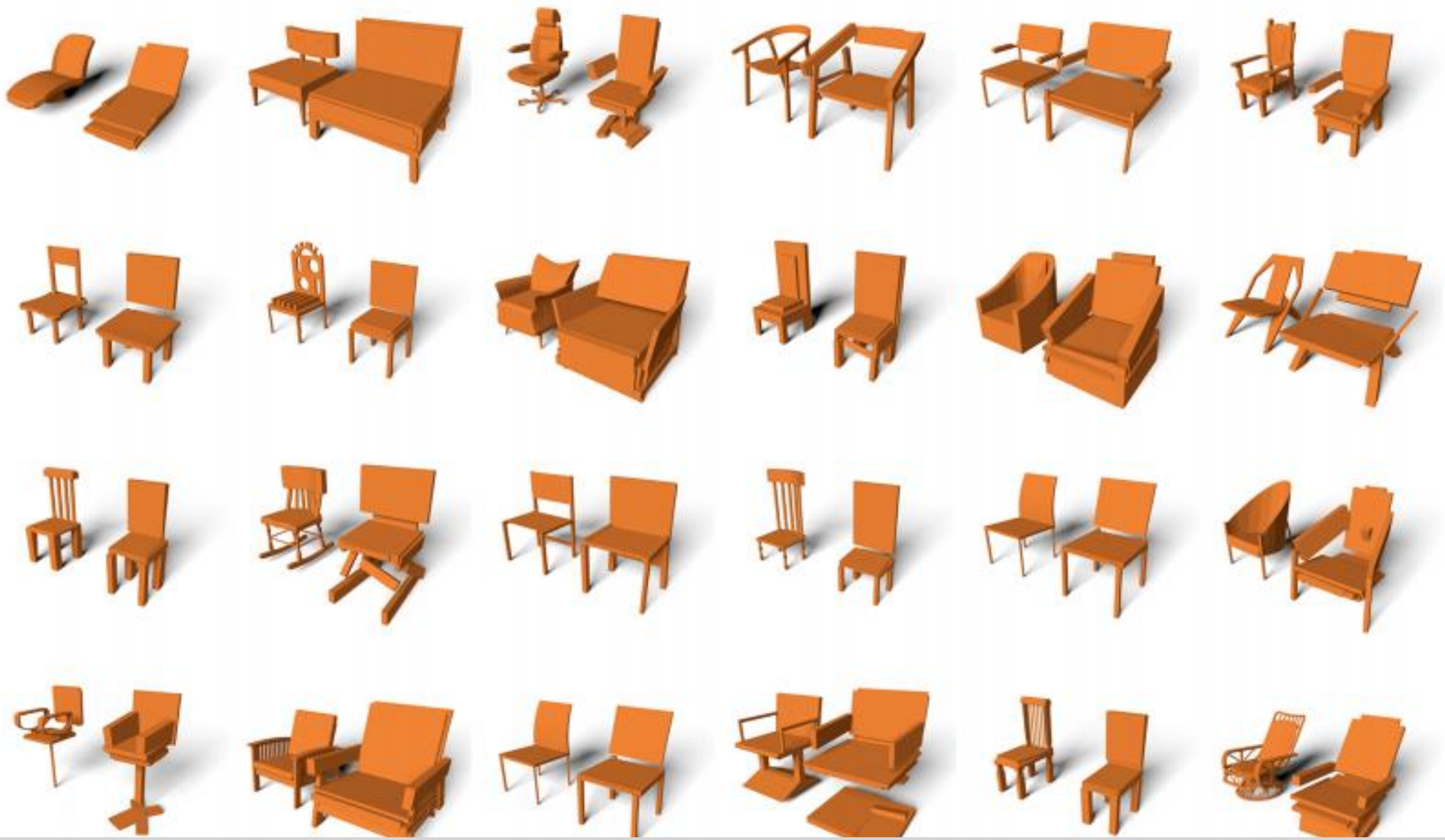


Results



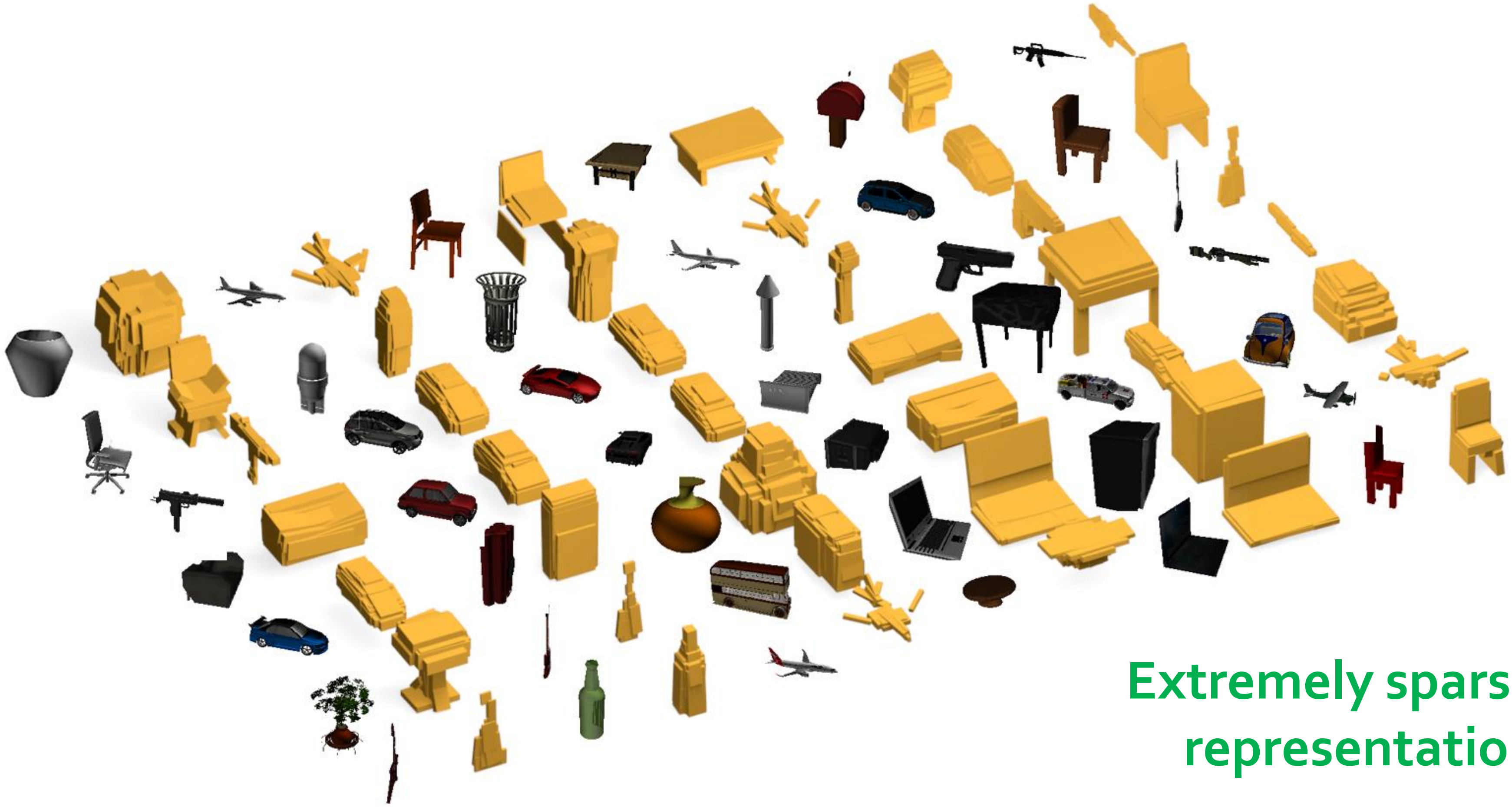
Interpolation

Results



Self-Supervised Shape Abstraction

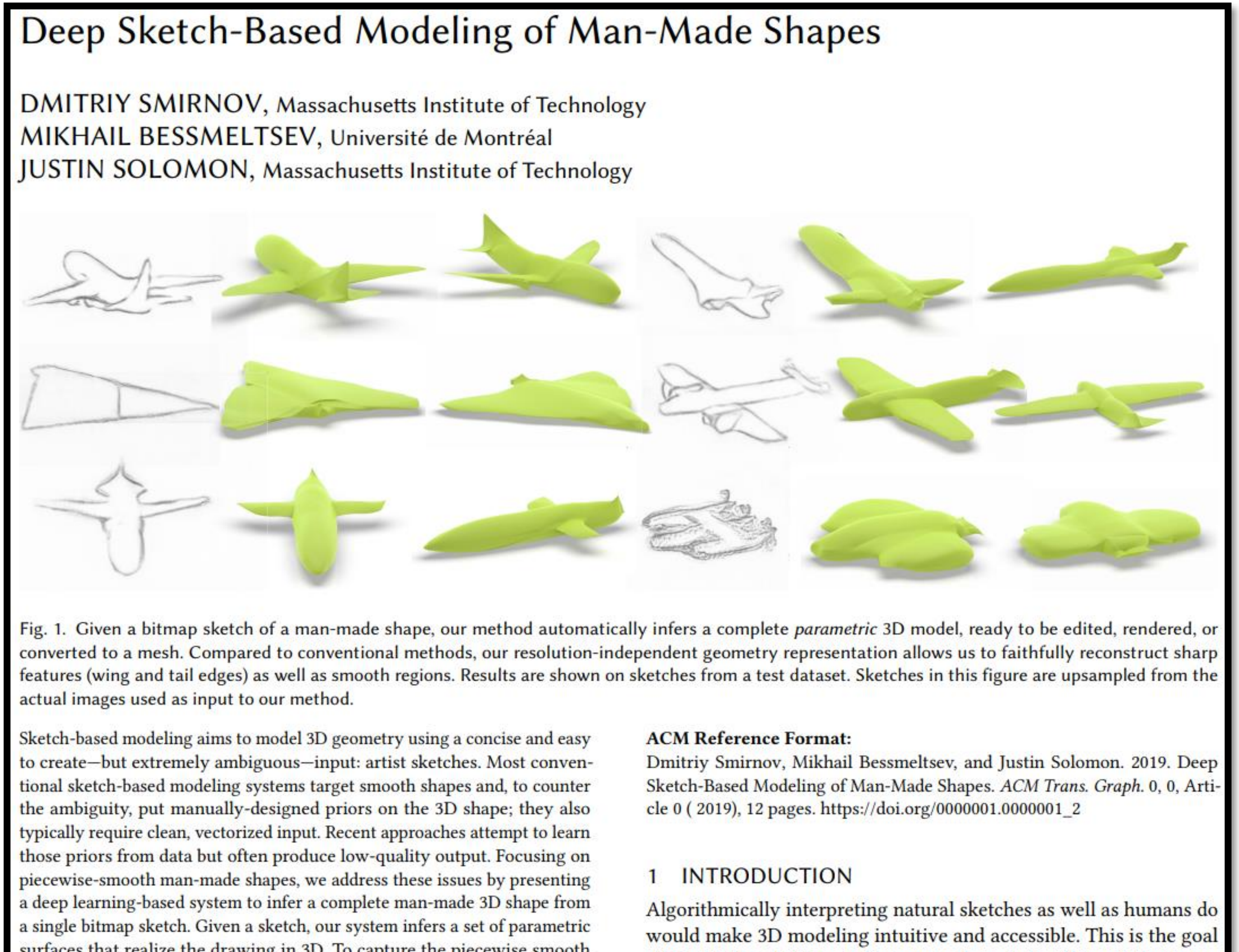
Results



Extremely sparse representation

Image-based 3D abstraction

Forthcoming Work



Deep Sketch-Based Modeling of Man-Made Shapes

DMITRIY SMIRNOV, Massachusetts Institute of Technology
MIKHAIL BESSMELTSEV, Université de Montréal
JUSTIN SOLOMON, Massachusetts Institute of Technology

Fig. 1. Given a bitmap sketch of a man-made shape, our method automatically infers a complete *parametric* 3D model, ready to be edited, rendered, or converted to a mesh. Compared to conventional methods, our resolution-independent geometry representation allows us to faithfully reconstruct sharp features (wing and tail edges) as well as smooth regions. Results are shown on sketches from a test dataset. Sketches in this figure are upsampled from the actual images used as input to our method.

Sketch-based modeling aims to model 3D geometry using a concise and easy to create—but extremely ambiguous—input: artist sketches. Most conventional sketch-based modeling systems target smooth shapes and, to counter the ambiguity, put manually-designed priors on the 3D shape; they also typically require clean, vectorized input. Recent approaches attempt to learn those priors from data but often produce low-quality output. Focusing on piecewise-smooth man-made shapes, we address these issues by presenting a deep learning-based system to infer a complete man-made 3D shape from a single bitmap sketch. Given a sketch, our system infers a set of parametric surfaces that realize the drawing in 3D. To capture the piecewise smooth

ACM Reference Format:
Dmitriy Smirnov, Mikhail Bessmeltsev, and Justin Solomon. 2019. Deep Sketch-Based Modeling of Man-Made Shapes. *ACM Trans. Graph.* 0, 0, Article 0 (2019), 12 pages. https://doi.org/0000001.0000001_2

1 INTRODUCTION

Algorithmically interpreting natural sketches as well as humans do would make 3D modeling intuitive and accessible. This is the goal

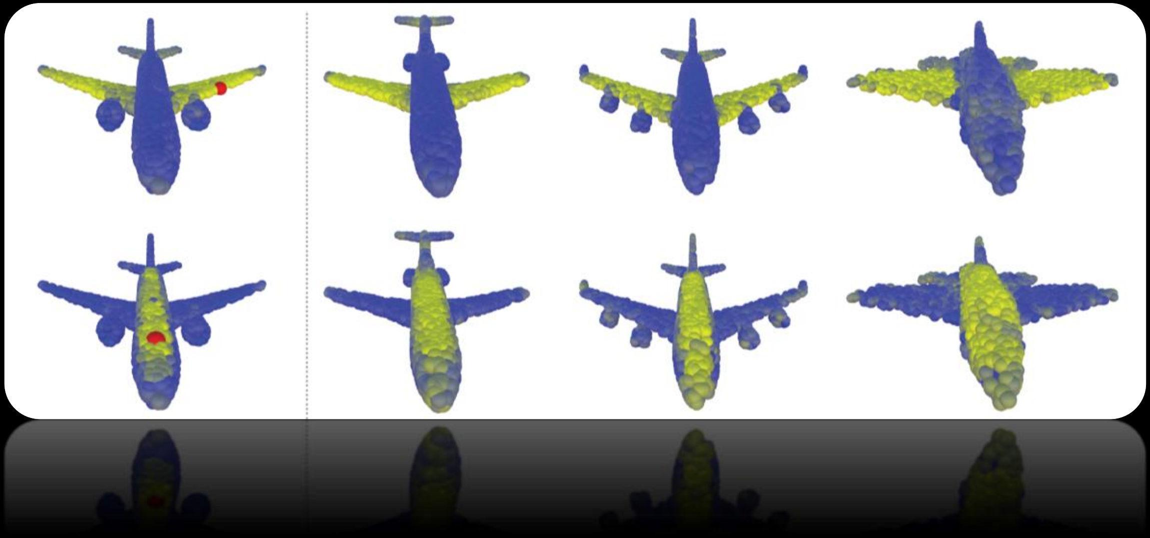
Take-Away

Learning from 3D data requires specialized, carefully-designed structures.

Many open problems!

Sensible Deep Learning

for 3D Data



Justin Solomon

MIT EECS