3D Object Classification via Spherical Projections

Zhangjie Cao[1], Qixing Huang[2], and Ramani Karthik[3]

[1]School of Software, Tsinghua University, China

[2]Department of Computer Science, University of Texas at Austin, USA

[3]School of Mechanical Engineering, Purdue University, USA

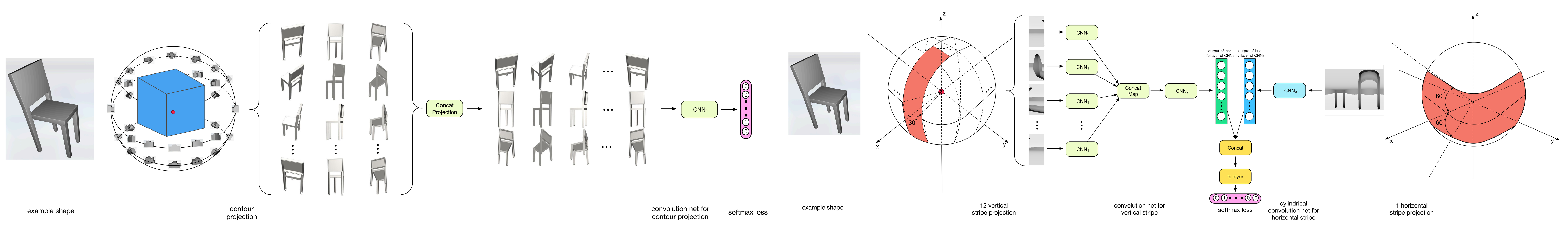


Figure 1: (left) contour-based projection and network; (right) depth-based projection and network

Summary

1. An novel approach for 3D object classification which unifies two key components:

- a spherical representation exploiting both depth variation and contour information which can capture geometric details and data dependencies across the entire object.
- deep neural networks incorporating large-scale labeled images for training to classify spherical representations of 3D objects.

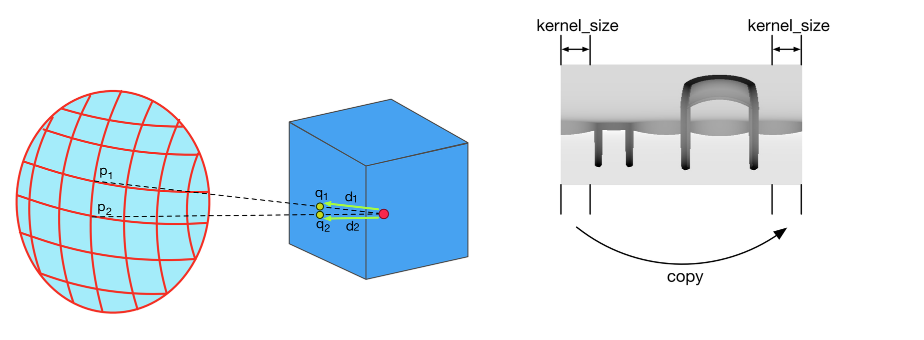
Challenges

1. Limitations of previous works:

- For 3D convolutional methods, the resolution of a 3D convolutional neural network is usually very coarse;
- For all 3D-based methods, the number of available training data is limited;
- For image-based methods, most do not capture the dependencies between views.

Our spherical projections combine key advantages of these two mainstream 3D classification methods.

Depth-based Projection



(a) Depth-based Projection Method (b) Cylindrical Depth-based Projection

Figure 2: Details on Depth-based Projection

- The left Figure indicates that each point is recorded as the distance to the first hitting point. Otherwise, the distance is set to be zero.
- The right Figure indicates the cylindrical convolution.

We first compute depth values for vertices of a semi-regular quad-mesh whose axis aligns with the longitude and latitude

$$(\cos(\theta_i)\cos(\phi_j), \cos(\theta_i)\sin(\phi_j), \sin(\theta_i))$$
$$\theta_i = \frac{180^\circ \cdot i}{m}, \phi_j = \frac{360^\circ \cdot j}{n}, \quad \begin{matrix} 0 \le i \le m-1 \\ 0 \le j \le n-1 \end{matrix} \tag{1}$$

Then, we generate the depth value of other points on the sphere by linear interpolation. Given a point with spherical coordinate (θ, ϕ), where $\theta_i \le \theta \le \theta_{i+1}, \phi_j \le \phi \le \phi_{j+1}$, its depth value is given by

$$\begin{aligned} d =& (1-t_{ij})\big((1-s_{ij})d_{ij} + s_{ij}d_{i,j+1}\big) \\ &+ t_{ij}\big((1-s_{ij})d_{i+1,j} + s_{ij}d_{i+1,j+1}\big). \end{aligned} \tag{2}$$

Finally, as in the Figure(right) of depth-projection, we proceed to generate cylindrical strips from the depth projection described above. We first use the strip covering the following area

$$(\cos(\theta_{i_h})\cos(\phi_{j_h}), \cos(\theta_{i_h})\sin(\phi_{j_h}), \sin(\theta_{i_h}))$$
$$\theta_{i_h} = \frac{120^\circ \cdot i_h}{m_h} + 30^\circ, \phi_{j_h} = \frac{360^\circ \cdot j_h}{n_h}, \quad \begin{matrix} 0 \le i_h \le m_h - 1, \\ 0 \le j_h \le n_h - 1. \end{matrix} \tag{3}$$

Since regions of high latitude suffer from severe distortion, we eliminate them by setting θ_{i_h} from 30° to 150°. To utilize information form high latitude regions for classification, we then use the following strip

$$\begin{aligned} &(\cos(\delta_{k_v})\cos(\theta_{i_v})\sin(\phi_{j_v}) - \sin(\delta_{k_v})\sin(\theta_{i_v}), \\ &\cos(\delta_{k_v})\sin(\theta_{i_v}) + \sin(\delta_{k_v})\cos(\theta_{i_v})\sin(\phi_{j_v}), \\ &\cos(\theta_{i_v})\cos(\phi_{j_v})) \\ &\theta_{i_v} = \frac{360^\circ \cdot i_v}{l_v m_v} + 90^\circ - \frac{180^\circ}{l_v}, \phi_{j_v} = \frac{180^\circ \cdot j_v}{n_v}, \delta_{k_v} = \frac{360^\circ \cdot j_v}{l_v} \\ &0 \le i_v \le m_v - 1, 0 \le j_v \le n_v - 1, 0 \le k_v \le l_v - 1 \end{aligned} \tag{4}$$

Contour-based Projection

As shown in Figure(left) of contour-based netwrok, contour-based projection shoots a 3x12 grid of images of the input object from 36 view points. The locations are $\phi_j = 0°, 30°, \cdots, 330°, \theta_i = -60°, 0°, 60°$. The up-right orientation of the camera always points to the north-pole. The viewing angle of each image is 45°. The resolution of each image is 224x224. In our experiments, we have varied the value of m and found that $m = 12$ provides a good trade-off between minimizing the number of views and ensuring that the resulting projections are approximately invariant to rotating the input object.

Experiment Setup

- **Datasets:** ModelNet40, ShapeNetCore
- **Parameter selection:** cross-validation by jointly assessing
- **Methods to compare with:** Image-based methods: MVCNN, MVCNN-MultiRes; 3D-based methods: 3D ShapeNets, Voxnet, Volumetric CNN, OctNet; combined methods: FusionNet. All of these methods use the upright orientation but do not use the front orientation.

Results

- Accuracy of our approaches and the various baseline methods on ModelNet40 and ShapeNetCore and two curated subsets.

Method	ModelNet40	ShapeNetCore	ModelNet40-SubI	ShapeNetCore-SubI
3D Shapenets	85.9	na	83.33	na
Voxnet	87.8	na	85.99	na
FusionNet	90.80	na	89.54	na
Volumetric CNN	89.9	na	88.65	na
MVCNN	92.31	88.93	91.22	88.64
MVCNN-MultiRes	93.8	90.01	92.60	90.00
OctNet	87.83	88.03	86.45	87.85
depth-base pattern	91.36	89.45	90.25	89.13
contour-based pattern	93.31	90.49	92.20	90.80
overall pattern	94.24	91.00	93.09	91.22

Classwise Comparison

- Accuracy of Each Class For Different Projection on ModelNet40

Class Name	Depth-Based	Contour-Based	MVCNN	Class Name	Depth-Based	Contour-Based	MVCNN
bowl	100.00	95.00	85.00	stool	75.00	75.00	75.00
bookshelf	99.00	99.00	94.00	tent	95.00	95.00	95.00
cone	100.00	100.00	95.00	toilet	100.00	100.00	100.00
table	89.00	84.00	84.00	xbox	80.00	80.00	80.00
vase	82.00	85.00	77.00	car	99.00	100.00	100.00
tv_stand	85.00	90.00	81.00	guitar	98.00	100.00	99.00
dresser	89.53	89.53	86.05	monitor	97.00	99.00	99.00
bottle	98.00	96.00	96.00	plant	85.71	89.80	87.76
sofa	98.00	99.00	97.00	range_hood	93.00	97.00	96.00
airplane	100.00	100.00	100.00	night_stand	75.58	86.05	80.23
bathtub	94.00	96.00	94.00	sink	85.00	85.00	90.00
bed	100.00	100.00	100.00	piano	91.00	96.00	97.00
bench	80.00	80.00	80.00	mantel	93.00	97.00	100.00
chair	98.00	99.00	98.00	curtain	85.00	95.00	95.00
desk	86.05	87.21	86.05	lamp	75.00	80.00	85.00
door	100.00	100.00	100.00	cup	55.00	80.00	70.00
glass_box	97.00	97.00	97.00	flower_pot	15.00	15.00	30.00
keyboard	100.00	100.00	100.00	wardrobe	65.00	90.00	90.00
laptop	100.00	100.00	100.00	radio	65.00	95.00	95.00
person	100.00	95.00	100.00	stairs	70.00	100.00	100.00

Running Time Analysis

- Running Time of All the methods on ModelNet40 and ShapeNetCore

Method	ModelNet40			ShapeNetCore		
	Rendering	Inference	Training	Rendering	Inference	Training
Depth	0.92s	0.043s	252m	1.06s	0.043	272m
Contour	1.17s	0.418s	371m	1.28s	0.418s	442m

Effect of Pre-training

- Accuracy Before and After Pre-training on ModelNet40

Method	Before Pre-training		After Pre-training	
	Accuracy (class)	Accuracy (instance)	Accuracy (class)	Accuracy (instance)
MVCNN	82.15	87.15	90.35	92.31
MVCNN-MultiRes	88.12	91.20	91.40	93.80
depth-base pattern	80.44	86.09	87.32	91.36
contour-based pattern	88.33	91.48	91.16	93.31
overall pattern	88.53	91.77	91.56	94.24

- Accuracy Before and After Pre-training on ShapeNetCore

Method	Before Pre-training		After Pre-training	
	Accuracy (class)	Accuracy (instance)	Accuracy (class)	Accuracy (instance)
MVCNN	67.84	84.55	78.79	88.93
MVCNN-MultiRes	75.34	88.4	79.01	90.01
depth-base pattern	70.55	85.15	78.84	89.45
contour-based pattern	74.52	88.54	79.38	90.49
overall pattern	75.60	88.87	80.38	91.00

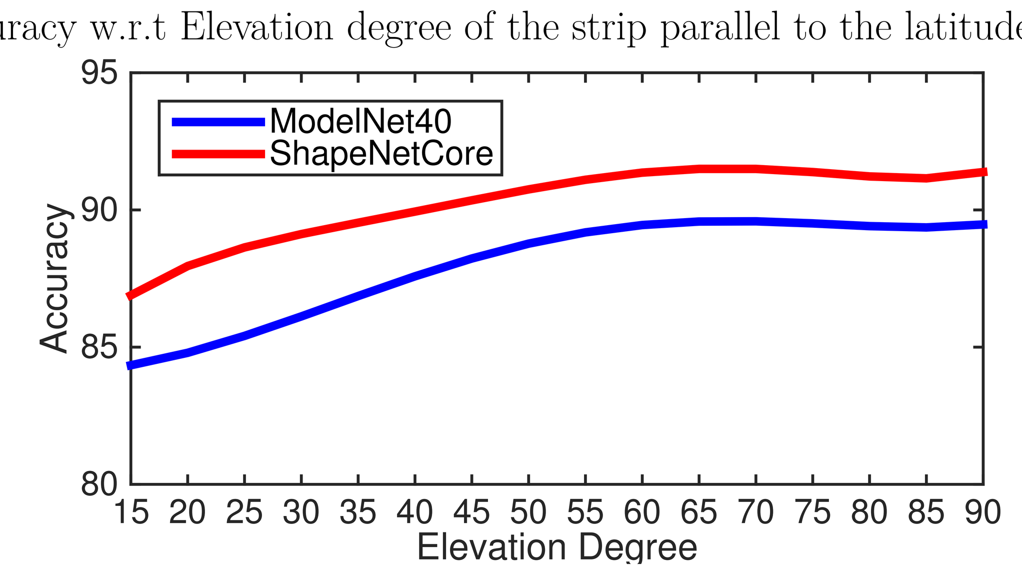
Number of Views

- Accuracy w.r.t Number of Views for Depth and Contour Pattern on ModelNet40 and ShapeNetCore

Pattern	Number of Views	ModelNet40	ShapeNetCore
depth-based	6	90.87	88.95
	24	91.02	89.23
	12	91.36	89.45
contour-based	6	92.26	89.23
	24	93.12	90.36
	12	93.31	90.49

Elevation Degree Variation

- Accuracy w.r.t Elevation degree of the strip parallel to the latitude



Contact Information

- Web: `https://www.cs.utexas.edu/~huangqx/`
- Email: `huangqx@cs.utexas.edu`

THE UNIVERSITY OF TEXAS AT AUSTIN · PRAESIDIUM · DISCIPLINA · CIVITATIS

3DV 2017