Spherical CNNs

Arjun Guru & Claudia Zhu

STAT 991 | Professor Dobriban

Why We Need Spherical CNNs

- ICLR 2018 Best Paper Award!!
- Convolutional Neural Networks with 2D convolutions
  - Vision tasks on 2D images
- What about 3D circular cameras and more general 3D data?
  - Ex. self-driving cars, drones, maps, temperature, wind, etc.

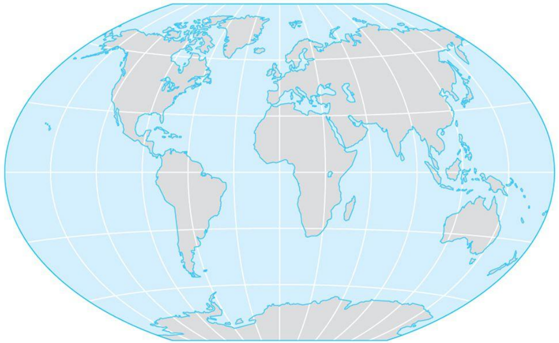
Notable Past Work

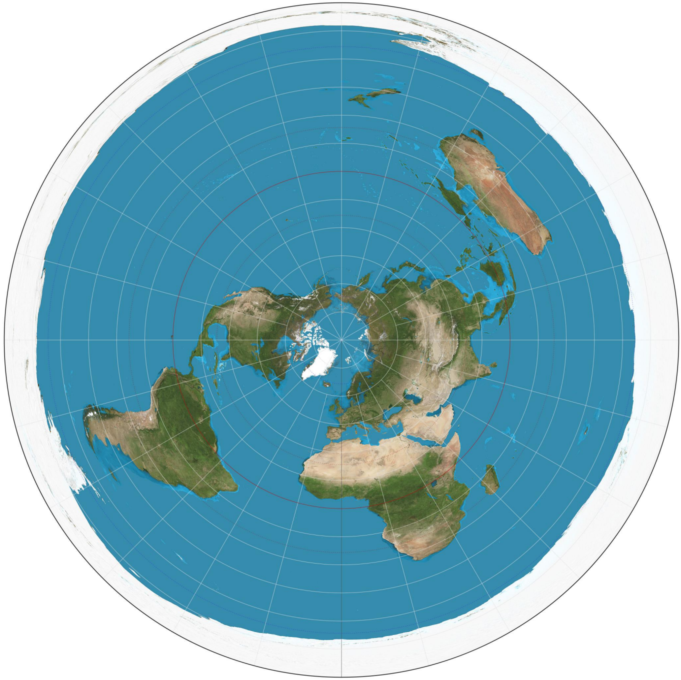
Generalizations for larger groups of symmetries (Cohen and Welling, Gens and Domingos)

Restricted to discrete rotations on planar images or permutations on point clouds

What are Spherical Images?

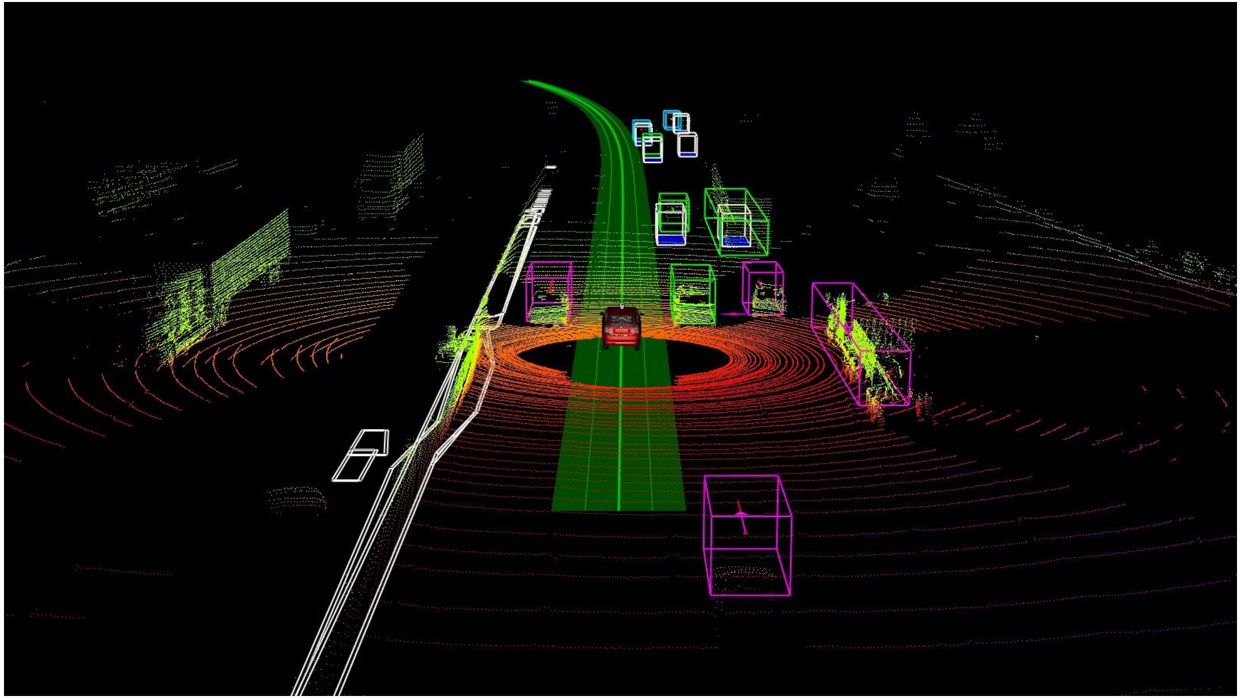
How do we teach a computer which one is Antarctica?





Point Clouds

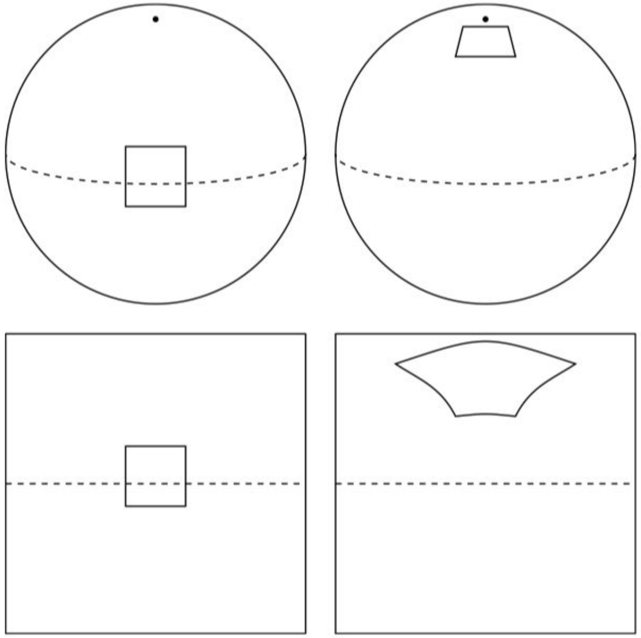
- A set of data points in space



Two Major Challenges:

No symmetrical grid (cannot consider single pixel)

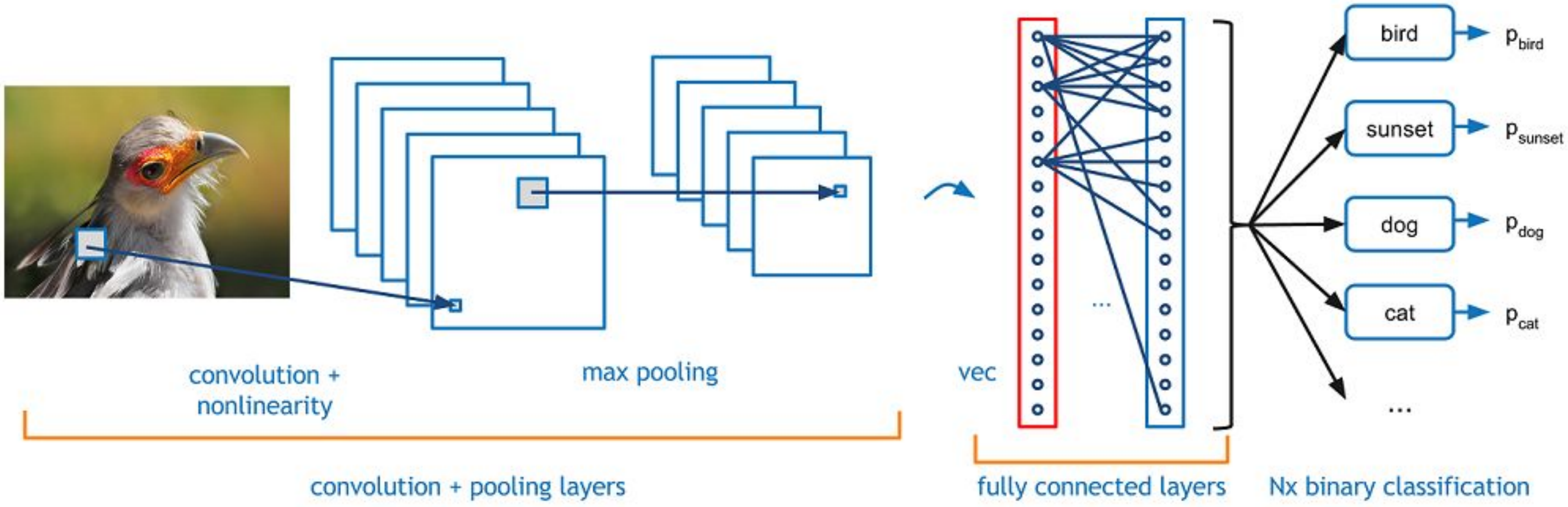
Computational Inefficiency O(n^6) SO(3) is 3D manifold



Contributions of the paper

1. Theory of Spherical CNNs
2. Implementation using Fast Fourier Transforms
3. Empirical Support

CNNs


convolution +
nonlinearity
max pooling
convolution + pooling layers
vec
...
fully connected layers
bird
sunset
dog
cat
...
p_{bird}
p_{sunset}
p_{dog}
p_{cat}
Nx binary classification

Theory of Spherical CNNs

Building Blocks

The Unit Sphere

S^2 can be defined as the set of points $x \in \mathbb{R}^3$ with norm 1

parameterized by spherical coordinates $\varphi \in [0, 2\pi]$ and $\theta \in [0, \pi]$

Spherical Signals

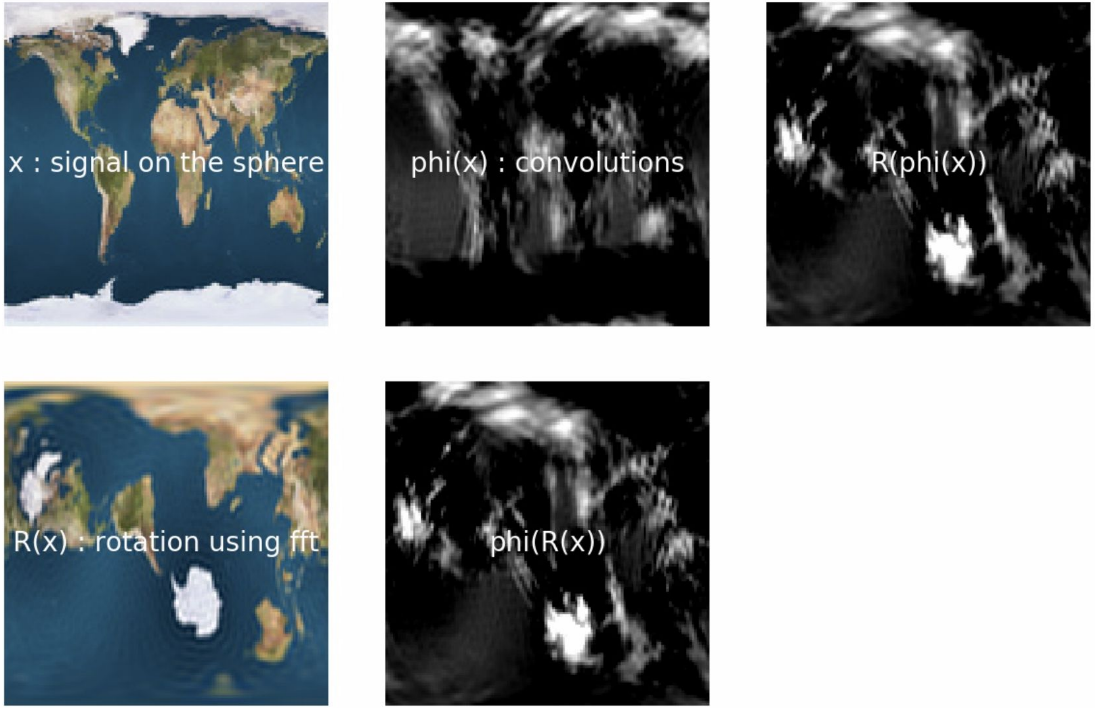
$$f : S^2 \to \mathbb{R}^K$$

$$x = x_0 + r \sin\theta \ \cos\varphi$$
$$y = y_0 + r \sin\theta \ \sin\varphi$$
$$z = z_0 + r \cos\theta$$

Rotation Group

$\mathrm{SO}(3)$ group of all rotations in 3D

Functions Example



Rotation Group Properties

First verify that this is a group

Represented as 3 x 3 matrices

Preserves Distance (an isometry)

Preserves Orientation (det = +1)

Represent points of sphere as unit vectors

Parametrize as:

ZYZ-Euler angles $\alpha \in [0, 2\pi]$, $\beta \in [0, \pi]$, and $\gamma \in [0, 2\pi]$

Rotations

Rotation Operator: L_R and filter f we have $L_R f$

$$[L_R f](x) = f(R^{-1}x)$$

Inner Product

$$\langle \psi, f \rangle = \int_{S^2} \sum_{k=1}^{K} \psi_k(x) f_k(x) dx$$

Inner Product

$$\int_{S^2} f(Rx)dx = \int_{S^2} f(x)dx$$

Volume under spherical heightmap does not change when rotated

Inverse of Rotation Operator is adjoint to Rotation Operator

Rotation Operator is unitary

Properties of Unitary Operators:

- Preserves dot product
- Normal
- det = +1 (preserves orientation)
- $(Ax, y) = (x, By)$

Inverse of Rotation Operator is Adjoint to Rotation Operator

$$\begin{aligned}\langle L_R\psi, f\rangle &= \int_{S^2}\sum_{k=1}^{K}\psi_k(R^{-1}x)f_k(x)dx\\ &= \int_{S^2}\sum_{k=1}^{K}\psi_k(x)f_k(Rx)dx\\ &= \langle\psi, L_{R^{-1}}f\rangle.\end{aligned}$$

2D Correlation for Spherical Correlation

Correlation for Translations in Plane

Value of the output feature map at translation is computed as an inner product between the input feature map and a filter shifted by $x \in \mathbb{Z}^2$

Correlation for Rotations in Sphere

Value of the output feature map evaluated at rotation R is computed as an inner product between the input feature map and a filter rotated by $R \in \mathrm{SO}(3)$

Spherical Correlation

$$[\psi \star f](R) = \langle L_R \psi, f \rangle = \int_{S^2} \sum_{k=1}^{K} \psi_k(R^{-1}x) f_k(x) dx$$

Spherical Convolution (Driscoll and Healy)

Where *n* is the north pole

$$[f * \psi](x) = \int_{\mathrm{SO}(3)} f(Rn) \psi(R^{-1}x) dR$$

Rotation Group Correlation

- We talked about how we can apply a spherical convolution to a spherical image
- The output of a spherical convolution is a function on SO(3)
- What do subsequent spherical convolution layers do then?
- We can show that a spherical convolution can also be applied to a function on SO(3)
    - (In addition to functions on the unit sphere)

$$[\psi \star f](R) = \langle L_R \psi, f \rangle = \int_{\mathrm{SO}(3)} \sum_{k=1}^{K} \psi_k(R^{-1}Q) f_k(Q) dQ.$$

Equivariance

- Why is equivariance important?
- Consider a 2D convolution operation and where in the output we see a maximum activation
  - If the image is translated, max activation location **is also translated** (but not transformed in any other way)
- The definition of equivariance is as follows:
  - Consider a group **G** of operations
  - A function *f* is equivariant under **G** if for a given transformation g in **G**:
    - *f*(g(<input>) = g’(f(<input>))
    - In words: when a function is equivariant under a group of transformations:
      - The function applied to an input transformed by an element of the group is equal to some other transformation in the group, applied to the output of the function with an unmodified input

Showing Equivariance

$$
\begin{aligned}
[\psi \star [L_Q f]](R) &= \int_{S^2} \psi(R^{-1}x) f(Q^{-1}x) dx \\
&= \int_{S^2} \psi(R^{-1}Qx) f(x) dx \\
&= \int_{S^2} \psi((Q^{-1}R)^{-1}x) f(x) dx \\
&= [\psi \star f](Q^{-1}R) \\
&= [L_Q[\psi \star f]](R)
\end{aligned}
$$

By Linear Adjointness

Implementation of Spherical CNNs

Efficient Implementations of Spherical Convolutions

- Naive implementation of spherical convolution is O(n^6)
- Convolutions can be computed efficiently using Fast Fourier Transforms
- For more details on this, please refer to the original paper:
    - https://arxiv.org/pdf/1801.10130.pdf

Rotated MNIST

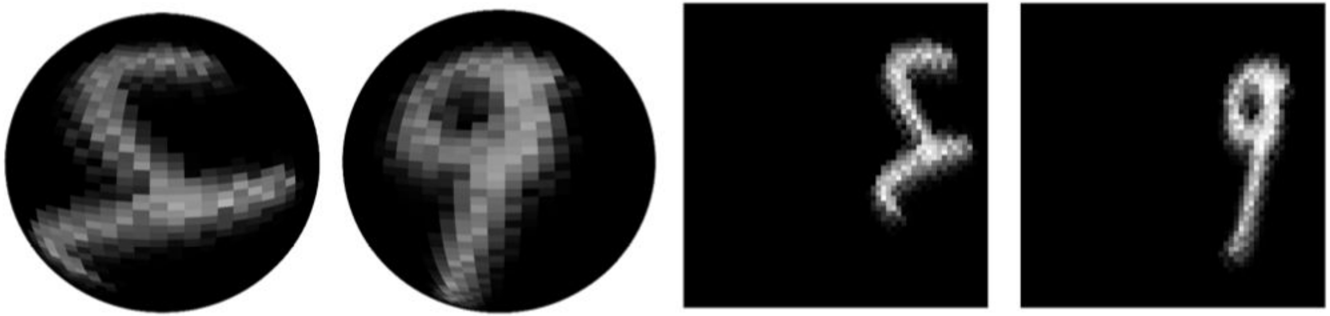


Figure 4: Two MNIST digits projected onto the sphere using stereographic projection. Mapping back to the plane results in non-linear distortions.

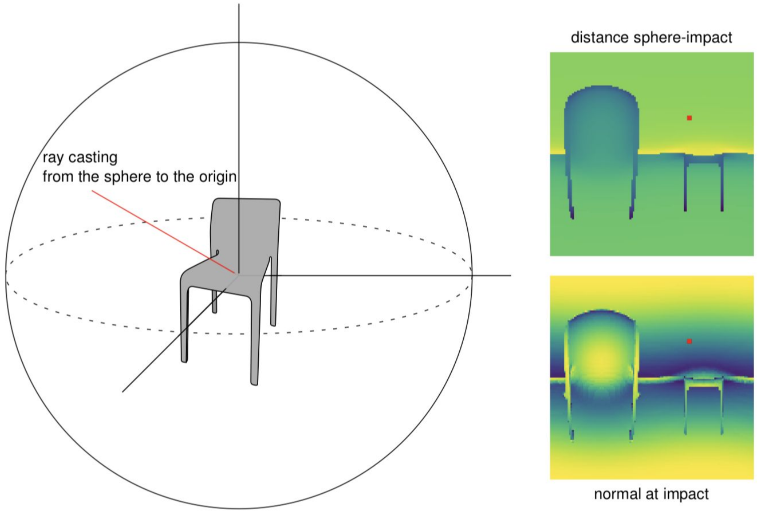
Results: Rotated MNIST

- Goal: evaluate generalization performance of network after rotations of input
- Dataset: MNIST digits projected onto a sphere
  - First instance: each digit is projected onto northern hemisphere (NR-Non-Rotated)
  - Second instance: same as above, but sphere is then randomly rotated (R-Rotated)
- Training and Testing Accuracy on NR/NR vs. NR/R vs. R/R, respectively (below):

	NR / NR	R / R	NR / R
planar	0.98	0.23	0.11
spherical	0.96	0.95	0.94

Representing Mesh Data as a Spherical Image

- Ray Tracing
  - Draw a ray from sphere back to origin, and note first intersection with object
- 3 types of information gathered from intersection:
  - Ray Length
  - Cos of Surface Angle
  - Sin of Surface Angle



Results on Recognition of 3D Shapes

- We just discussed a spherical representation of 3D meshes
- The Spherical CNN model was compared against models highly specialized for the SHREC17 task (3D object classification)
- Spherical CNN was in 2nd place in most of these metrics while not being highly specialized to the particular dataset, unlike these other top performing models

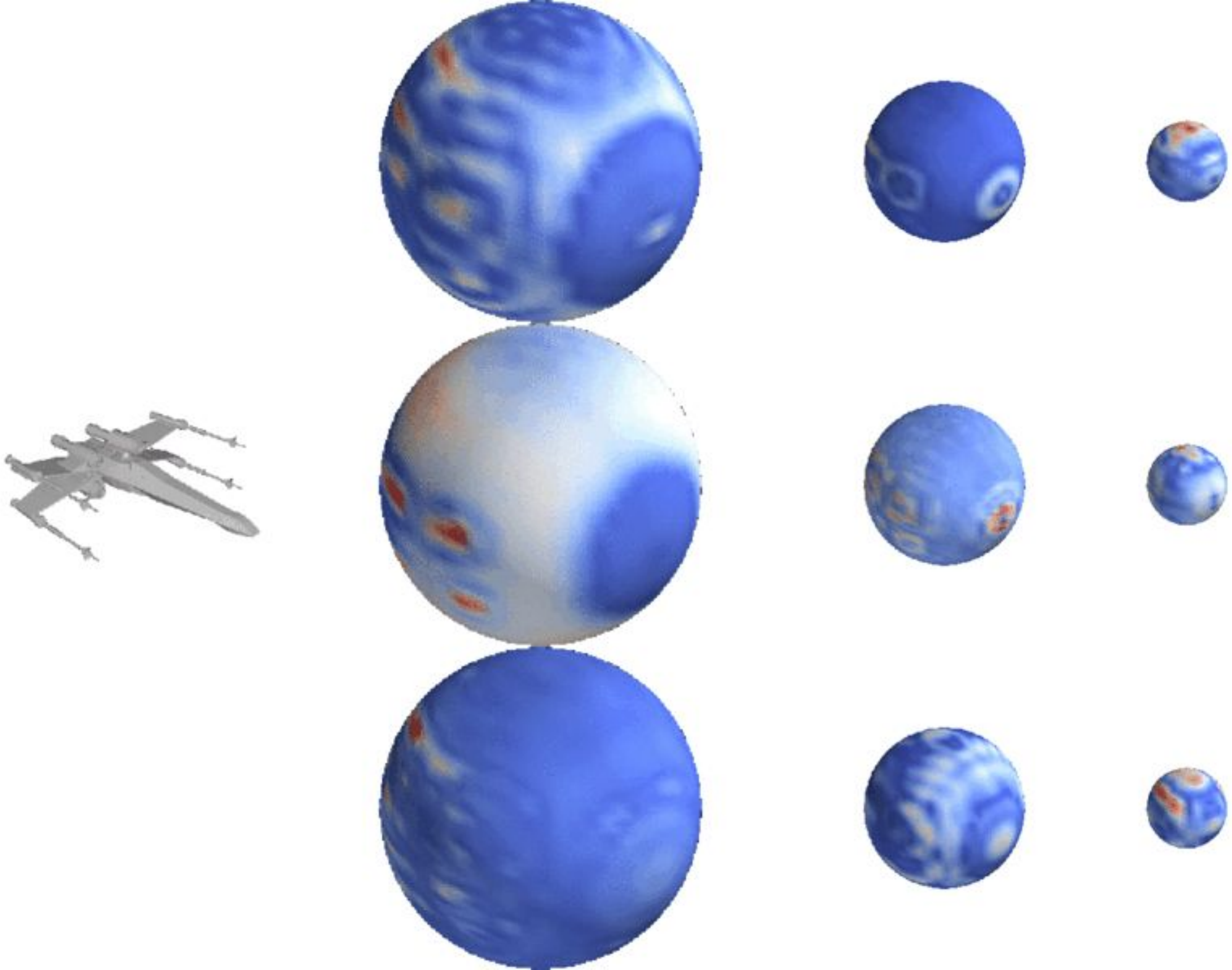
Method	P@N	R@N	F1@N	mAP	NDCG
Tatsuma_ReVGG	0.705	0.769	0.719	0.696	0.783
Furuya_DLAN	0.814	0.683	0.706	0.656	0.754
SHREC16-Bai_GIFT	0.678	0.667	0.661	0.607	0.735
Deng_CM-VGG5-6DB	0.412	0.706	0.472	0.524	0.624
Ours	0.701 (3rd)	0.711 (2nd)	0.699 (3rd)	0.676 (2nd)	0.756 (2nd)

Table 2: Results and best competing methods for the SHREC17 competition.

Demos!!!

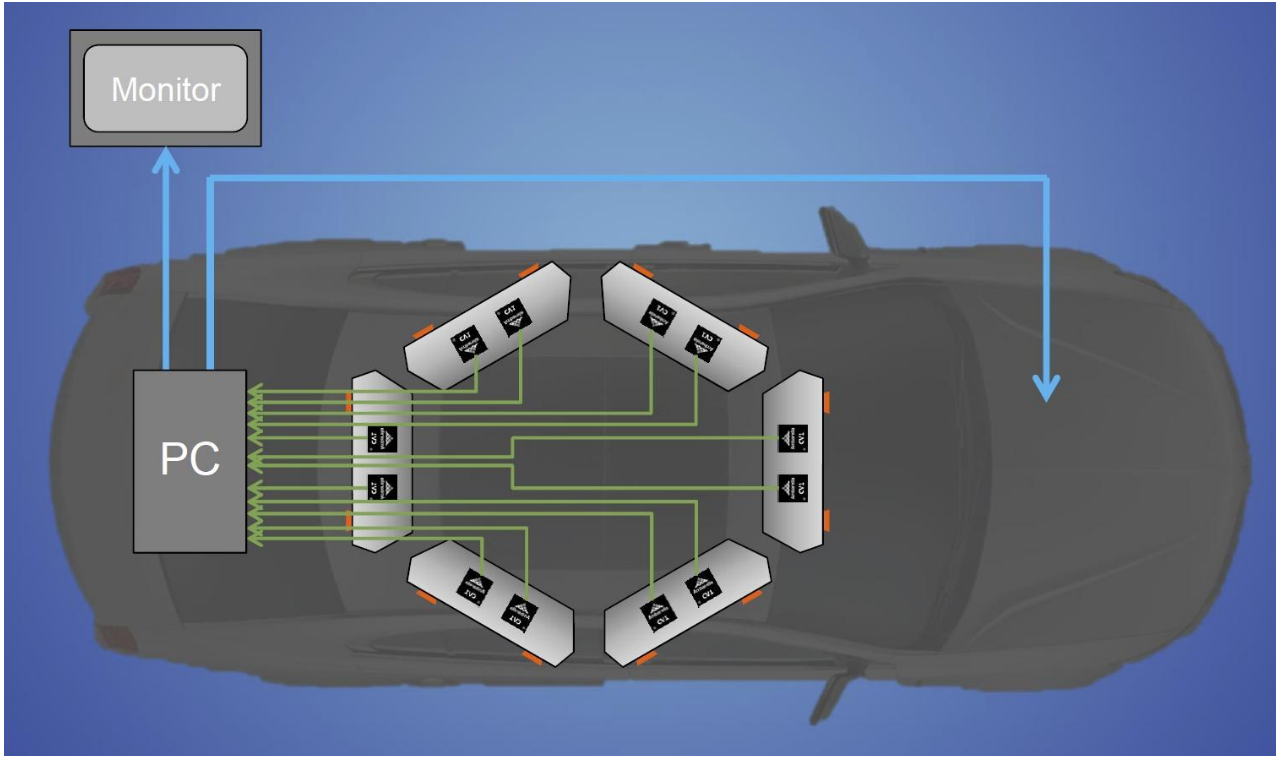
https://github.com/daniilidis-group/spherical-cnn

https://github.com/jonas-koehler/s2cnn



Future Applications

Self Driving Cars



More Spherical Data

Omnidirectional Images Example

References

T.S. Cohen, M. Geiger, J. Koehler, M. Welling, Spherical CNNs. ICLR 2018

Professor Dobriban's Lecture Notes Section 2.6

Spherical CNN Demo -- https://github.com/daniilidis-group/spherical-cnn

Spherical CNN Demo -- https://github.com/jonas-koehler/s2cnn