Deep Learning 3D Shapes Using Alt-Az Anisotropic 2-Sphere Convolution

Anonymous authors
Paper under double-blind review

Abstract

The ground-breaking performance obtained by deep convolutional neural networks (CNNs) for image processing tasks is inspiring research efforts attempting to extend it for 3D geometric tasks. One of the main challenges in applying CNNs to 3D shape analysis is how to define a natural convolution operator on non-Euclidean surfaces. In this paper, we present a method for applying deep learning to 3D surfaces using their spherical descriptors and alt-az anisotropic convolution on 2-sphere. A cascade set of geodesic disk filters rotate on the 2-sphere and collect multi-level spherical patterns to extract non-trivial features for various 3D shape analysis tasks. We demonstrate theoretically and experimentally that our proposed method has the possibility to bridge the gap between 2D images and 3D shapes with the desired rotation equivariance/invariance, and its effectiveness is evaluated in applications of non-rigid/rigid shape classification and shape retrieval.

1 Introduction

A recent research effort in computer vision and geometric processing communities is towards replicating the incredible success of deep convolutional neural networks (CNNs) from the image analysis to 3D shape analysis. A straightforward extension is to treat a 3D shape as a voxel grid (Wu et al. (2015); Maturana & Scherer (2015); Song & Xiao (2016); Wang et al. (2017); Riegler et al. (2016).) Alternative methods include encoding a 3D shape as a collection of 2D renderings from multiple cameras (Qi et al. (2016); Su et al. (2015); Bai et al. (2016),) or projecting a 3D object onto geometric entities which can be flattened as 2D images (Shi et al. (2015); Cao et al. (2017); Sfikas et al. (2018).) All these methods convert a 3D shape into an Euclidean grid structure which supports shift-equivariance/invariance, such that conventional CNNs can work out-of-the-box.

Although embedded in \( \mathbb{R}^3 \), 3D shapes are typically represented as manifold surfaces. Recent research has particularly focused on convolutional networks for non-Euclidean domains such as manifolds or graphs. One of the main difficulties of adopting CNNs and similar methods in these non-Euclidean domains is the lack of shift-invariance on surfaces or graphs (see Masci et al. (2015).) Our motivation comes from the representation of 3D shapes as functions on spheres. We transfer the problem of manifold surface convolution into spherical convolution with the primary benefit of rotation invariance. Although shift-invariance is hard to achieve on general surfaces, by replacing filter translations with filter rotations, rotation equivariance/invariance can be obtained on the 2-sphere. Furthermore, spherical descriptors of 3D shapes are compact and rotation equivariant requiring a network of lower capacity, compared to voxel or multi-view representations. In this work, we are primarily interested in analyzing 3D geometric data using a specific type of spherical convolution either for classification or retrieval tasks.

2 Related Work

Surface convolution One approach to shift-invariance on surfaces is using re-parameterization methods. Geodesic CNN (Masci et al. (2015); Bronstein et al. (2017)) uses local geodesic polar coordinates to parameterize a surface patch locally around a point. An angular max-pooling layer is proposed to account for the filter’s local rotational degree of freedom. Spherical parameterization methods (Peng & Timalsena (2016); Praun & Hoppe (2003); Gu et al. (2004)) map a genus-0 3D
Figure 1: An example of our alt-az anisotropic spherical convolution neural network (a³S-CNN) applied for a non-rigid shape classification problem.

shape onto a sphere bijectively which provides a global framework for the spherical convolution. Sinha et al. (2016; 2017) transfer a genus-0 3D shape into a parameterized spherical image, and then flatten it into a planar geometry image. Data augmentation is necessary for geometry images in order to account for inconsistent cut positions and orientations. Toric covers (Maron et al. (2017)) is a seamless representation which stitches four copies of a genus-0 surface and globally maps them onto a planar flat torus. Spectral methods perform convolution on the spectral domain using the graph Laplacian and its eigen space decomposition (Yi et al. (2016); Bruna et al. (2013)). This method can efficiently address the shift-invariance problem, however, it suffers the difficulty with cross-shape learning since the spectral decomposition of each shape can be inconsistent.

Spherical convolution Spherical representation of 3D shapes have been used for shape matching (Kazhdan et al. (2003); Frome et al. (2004); Makadia & Daniilidis (2010)), remeshing (Praun & Hoppe (2003)), medical imaging (Shen & Makedon (2006)) and other tasks before the deep learning era. Recently, researchers have started to explore deep spherical convolutional neural networks for tasks such as molecular modeling (Boomsma & Frellsen (2017)), omnidirectional vision (Su & Grauman (2017)) and 3D shape recognition (Cohen et al. (2018); Esteves et al. (2018)). Su & Grauman (2017) discretized a spherical image using a lat-lon grid (see Fig.3(a)) and flattened it through equirectangular projection. A variable filter size is proposed to compensate for the imbalanced sampling along longitudinal direction. In Boomsma & Frellsen (2017), a cubed-sphere grid (see Fig.3(b)) is investigated in addition to a lat-lon grid, to achieve relatively more uniform grid on spheres. The work of Cohen et al. (2018) generalizes the spherical convolution with the full three rotational degrees of freedom in the 3D space, and it maps a spherical image to features on SO(3) using generalized Fourier transform. A similar work is done in Esteves et al. (2018) with azimuthally symmetric filters.

In this paper, we propose an alt-az anisotropic spherical convolutional neural network (or a³SCNN for short) for various rigid and non-rigid shape analysis tasks. Fig. 1 gives an overview of our method. A 3D shape is represented as a set of spherical images using spherical parameterization (for non-rigid shapes) or spherical projection (for rigid shapes, not shown in the figure). An icosahedron based spherical grid is used as the discrete representation of the spherical images. The convolution is applied directly on the spherical representation of the shape using a geodesic disc shape of filter. The proposed deep a³SCNN has multiple sequential convolutions followed by a nonlinearity such as ReLU and Spherical (Max or average) Pooling, all conducted on the spherical domain. Output is a set of spherical images which capture high-level shape feature descriptors. Following are the main contributions of our paper:

1. theoretical analysis of the relationship between various definition of convolutions for functions defined on the 2-sphere and a novel convolutional neural network using alt-az anisotropic spherical convolutions that emulates most aspects of convolutional networks in \( \mathbb{R}^2 \);
(2) an efficient geodesic grid data structure to support fast computation of the spherical convolution with locally-supported geodesic disc filters;
(3) an empirical demonstration of the utility of $a^3$SCNN with 3D shape learning problems.

3 ALT-AZ CONVOLUTION ON 2-SPHERE

We now describe the main contribution of our paper, i.e., the alt-az convolution on 2-sphere. We first describe the notations and then describe the alt-az convolution on context of other convolutions for spherical functions.

3.1 NOTATIONS AND PRELIMINARIES

2-sphere or unit sphere $S^2$ can be regarded as the set of points $u \in \mathbb{R}^3$ with norm one. The 2-sphere is a 2-manifold on which any point $\hat{u}$ is a unit vector. The $\hat{u}$ can be parametrized by spherical coordinates $(\theta, \phi) \in [0,\pi] \times [0,2\pi]$ such that $\hat{u}(\theta,\phi) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$. A regular region $r$ on $S^2$ has a positive area $A = \int_r ds(\hat{u})$, where $ds(\hat{u}) = \sin \theta d\theta d\phi$.

A special region on the 2-sphere is called polar cap region $R_{\theta_0}$, around the north pole, $\hat{u}(0,0,1)$, which is azimuthally symmetric and is parameterized by a maximum colatitude angle $\theta_0$:

$$R_{\theta_0} \triangleq \{(\theta, \phi) : 0 \leq \theta \leq \theta_0, 0 \leq \phi \leq 2\pi\}. \quad (1)$$

3D Rotations The set of rotations in three dimensions is called “special orthogonal group” $SO(3)$. $SO(3)$ is a 3-manifold on which any rotation $R \in SO(3)$ can be represented as a $3 \times 3$ matrix. Each rotation $R$ is associated with three independent parameters, we use the right hand rule zyz-Euler angles $\varphi \in [0,2\pi], \theta \in [0,\pi], \text{ and } \omega \in [0,2\pi]$, i.e.

$$R \equiv R^{\varphi\theta\omega}(z) \triangleq R^\varphi(z) R^\theta(y) R^\omega(z) \quad (2)$$

If we fix the third rotation angle $\omega$ to zero, $SO(3)$ is reduced into a subgroup termed alt-az rotation group. Any rotation $R$ in this group can be described as:

$$R \equiv R^{\varphi\theta}(z) \triangleq R^\varphi(z) R^\theta(y) \quad (3)$$

The group $SO(3)$ is used to describe all possible orientations of an object in 3D space, whereas the alt-az rotation group is generally used to describe the upright orientation of an object.

Rotation operator We define the effect of general rotation on spherical functions as an operator $D_R(\varphi, \theta, \omega)$ which corresponds to the rotation matrix $R$ defined in Eqn. (2). The effect of $D_R(\varphi, \theta, \omega)$ on the spherical image $f$ can be realized through an inverse rotation $R^{-1}$ of the coordinate system. That is,

$$<D_R(\varphi, \theta, \omega)f > = f(R^{-1}\hat{u}). \quad (4)$$

As shown in [Cohen et al., 2018], the rotation operator is unitary, i.e.

$$<D_R f, g> = <f, D_R^{-1} g>, \quad (5)$$

where $D_R^{-1} \triangleq D_R(-\varphi, -\theta, -\omega)$.

3.2 CONVOLUTION ON THE 2-SPHERE

The convolution operator in $n$ dimensional Euclidean space $\mathbb{R}^n$ is given by:

$$(h \otimes f)(x) \triangleq \int_{\mathbb{R}^n} h(x-y)f(y) dy, \quad x \in \mathbb{R}^n \quad (6)$$

The above equation is used as a reference to develop different notations of convolution on the 2-sphere.

Unlike conventional Euclidean domain signal, for spherical functions there is no consistent and well defined convolution operators. Two competing definitions exist in literature:
Type I: General anisotropic convolution: This convolution operator on 2-sphere tries to emulate the convolution in Euclidean spaces by replacing translations with full rotation in SO(3) and integrating over all possible rotations. This gives the most general definition of spherical convolution. Given a spherical filter \( h \) and spherical image \( f \) evaluated at a point \( \mathbf{u} \in \mathbb{S}^2 \), general anisotropic convolution on \( \mathbb{S}^2 \) is defined:

\[
h \odot f(\mathbf{R}) = g(\varphi, \vartheta, \omega) \triangleq \int_{\mathbb{S}^2} \sum_{k=0}^{K} (\mathcal{D}_R(\varphi, \vartheta, \omega)h)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}})
\]

(7)

Note that the output function \( g \) is not defined on the original \( \mathbb{S}^2 \). Instead, it is a function of three Euler angles \( (\varphi, \vartheta, \omega) \) and is therefore defined on the 3-manifold SO(3) (please see Cohen et al. (2018) for detail.)

Type II: Azimuthally isotropic convolution: This spherical convolution outputs a function defined on \( \mathbb{S}^2 \) using an azimuthally symmetric filter \( h_0(\mathbf{u}) \) (Esteves et al. [2018]; Driscoll & Healy [1994]):

\[
h_0(\hat{\mathbf{u}}) \triangleq \int_0^{\pi} \frac{1}{2\pi} (\mathcal{D}_{R_z}(\omega)h)(\hat{\mathbf{u}}) d\omega
\]

(8)

\[
h \odot f(\mathbf{R}) = g(\varphi, \vartheta) \triangleq \int_{\mathbb{S}^2} \sum_{k=0}^{K} (\mathcal{D}_{R_z}(\varphi) \circ \mathcal{D}_{R_y}(\vartheta) h_0)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}})
\]

(9)

Referring to Eqn. (9), we see that an arbitrary filter \( h \) is essentially transformed into a rationally symmetric filter \( h_0 \) through circular “averaging”. Type II spherical convolution zeros the contribution of angular variations from a filter, and hence, is considered restrictive for pattern matching purpose in spherical image processing.

Towards developing a spherical convolution which respects some important properties of convolutions defined in \( \mathbb{R}^2 \), we propose to use alt-az spherical convolution. In \( \mathbb{R}^2 \), the two spatial translations are isometric mappings and are directly convolved, whereas the isometry corresponding to a rotation in \( \mathbb{R}^2 \) is generally not convolved. The reasons for not using an rotation invariant filter in \( \mathbb{R}^2 \) are twofold. First, because local feature orientations are important cues for extracting higher level features, the rotation invariance should be considered in a global level rather than local level. Allowing filter to rotate introduces inconsistency in patching local features and thus, we expect rotation invariant filters to harm task performance. Second, from a computational viewpoint, it is much more efficient to learn global rotation of images using data augmentation and max pooling layers, as opposed to increasing dimensionality of filters and feature maps in each layer. Similarly, in the spherical domain, we believe that not all the isometries expressible in SO(3) are important from the perspective of spherical convolutions. The two degrees of freedom in alt-az rotation group are the direct analogs of two spatial translations in \( \mathbb{R}^2 \) (“shifting on the sphere”), and the third rotation \( \mathbf{R}_\omega \) should be fixed, emulating the filters with a fixed rotational degree of freedom in \( \mathbb{R}^2 \). Intuitively, we want to shift a spherical disc filter on the 2-sphere without self rotating. We now formally define our alt-az spherical convolutional operator.

Type III: alt-az anisotropic spherical convolution: Constraining the rotation of filter within the alt-az rotation group, the filter spans the altitude change by \( \vartheta \) and azimuth change by \( \varphi \), and is convolved with the signal. Mathematically, the alt-az convolution is defined as:

\[
(h \star f)(\mathbf{R}) = g(\varphi, \vartheta, 0) \triangleq \int_{\mathbb{S}^2} \sum_{k=0}^{K} (\mathcal{D}_R(\varphi, \vartheta, 0)h)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}).
\]

(10)

alt-az spherical operator has the following desirable properties:

- **Domain consistency**: It takes two functions in \( L^2(\mathbb{S}^2) \) and generates a function back in \( L^2(\mathbb{S}^2) \), such that cascaded layers of spherical convolutions can be utilized to exact hierarchical spherical patterns;

- **Alt-az rotation equivariance**: A layer \( \ell \) is rotation equivariant if \( \ell \circ \mathcal{D}_Q = \mathcal{D}_Q \circ \ell \). In our case, \( Q \) is a rotation in alt-az rotation group. We have,

\[
(h \star \mathcal{D}_Q f)(\mathbf{R}) = \mathcal{D}_Q (h \star f)(\mathbf{R})
\]

(11)
Figure 2: Rotation operators applied on a locally-supported kernel function. (a) An anisotropic kernel function $h$ defined on a polar cap; (b) applying an alt-az rotation $D_R(\pi/4, \pi/3, 0)$ to $h$; (c) applying a general rotation $D_R(\pi/4, \pi/3, \pi/2)$ to $h$; (d) applying a general rotation $D_R(\pi/4, \pi/3, \pi/2)$ to an azimuthally symmetric filter $h_0$.

Convolution with locally-supported filters Traditional CNNs are efficient due to the use of locally-supported filters and weight sharing. The metric on the Euclidean grid allows compactly supported filters, whose sizes are typically much smaller than the size of the input signal. On the 2-sphere, we propose to use locally-supported filters in the form of polar caps. Mathematically, a locally-supported filter is defined as a spacelimited spherical function belonging to the following subspace:

$$\mathcal{H}_{R_0}(S^2) \triangleq \{ h \in L^2(S^2) : h(\theta, \phi) = 0, \forall \theta > \theta_0 \},$$

(12)

where $R_0$ is the polar cap region on which the locally-supported filter is defined, and $\theta_0$ defines the size of a locally-supported filter. Fig. 2 shows a locally-supported filter undergoing different types of rotation.

4 NUMERICAL COMPUTATION OF ALT-AZ ANISOTROPIC CONVOLUTION

From a computational point of view, implementing the spherical convolution defined above in Eqns. (8-11) is difficult because it is not possible to uniformly discretize the surface of the sphere such that each sample point shares the same neighborhood. Therefore, a popular method of performing spherical convolution is to project the discretized spherical functions and filters onto the span of Wigner D functions for type I spherical convolution (see Cohen et al. (2018),) or spherical harmonics for type II spherical convolution (see Esteves et al. (2018).) They then perform the convolution in the Fourier domain via pointwise multiplications. The lack of locality support in the spherical Fourier transform inhibits us from using this method. Locally-supported filters belong to a subspace of spacelimited signals which is by nature infinite-dimensional. No non-trivial local filters can have a finite representation in the spectral domain. Esteves et al. (2018) use spectral smoothness to enforce a spatial decay in filters, and hence, achieve locality. However, the filters are still defined on the whole spherical domain which is memory inefficient. In this paper, we propose an alternate method which performs direct spherical convolution using geodesic grid discretization.

Figure 3: Different geodesic grids on the sphere. (a) a lat-lon grid, (b) a cubed-sphere grid and (c) icosahedron-sphere grid: from left to right, one-frequency, two-frequency, four-frequency and eight-frequency subdivisions.

Geodesic grid discretization Uniform geodesic grid cannot be achieved on the sphere except for the projections of five platonic polyhedra - tetrahedron, cube, octahedron, dodecahedron and icosahedron. These polyhedra can be further subdivided into different frequencies to obtain finer approximation of a sphere (Fig. 3(b) shows a subdivision of the projection of a cube on a sphere.)
Among the five platonic polyhedra, the icosahedron is most similar to the sphere. After the subdivision, the resulting triangulation has the least imbalance in area between its constituent triangle (Fig.3(c)). Most of the vertices have six direct neighbors except for the original 12 vertices of the icosahedron. This makes the icosahedron-based geodesic grid discretization most suitable for the discrete spherical convolution. We call this type of geodesic grid an icosahedron-sphere grid.

The total number of grid vertices are \( N = f^2 \times 10 + 2 \), where \( f \) is the subdivision frequency. Considering the structure of the icosahedron-sphere grid, in order to obtain a multi-level spherical feature map, the stride of a convolution or pooling layer has to be a multiple of \( 2^n \), where \( n \) is a positive integer. The stride is applied accordingly to the subdivision frequency \( f \). Fig.3(c) shows the icosahedron in different subdivision frequencies 1, 2, 4 and 8. A natural shape of the locally-supported filter correlating with the icosahedron-sphere grid, is a geodesic disc which can be discretized as a hexagonal grid of different ring sizes. Fig.4(c) shows two examples of such filters, one is of 1-ring size and the other is of size 2-ring. The same shape of geodesic disc and discretized hexagonal grid is used for local spherical pooling.

**Efficient data structure** The icosahedron-sphere grid data structure is self-sufficient to support spherical convolution, pooling and other CNN operators. However the linked data structure (vertex-edge-face and link data for topology) is not space efficient and is time consuming, in order to find the neighbors of a vertex and shift a filter on the sphere during the convolution. In this work, we use a rectilinear data structure to enable efficient spherical convolution and pooling. The icosahedron-based spherical mesh can be opened into 2D plane and represented as a grid structure as shown in Fig.4. The cut is along eleven edges of the icosahedron as shown in Fig.4(a) and (b). By rotating the \( u \) and \( v \) axes in Fig.4(b) into orthogonal axes, we obtain five rectangular 2D patches to store all the vertices of the icosahedron-sphere grid, as illustrated in Fig.4(c). This construction has two main advances: (1) within each patch, shifting filters on the 2-sphere is approximately equivariant to translation in \( u \) and \( v \) and (2) features on the geodesic grid can naturally be expressed using tensors, which means that the spherical convolution can be efficiently implemented on a GPU.

When implementing spherical convolutions and pooling operations for the icosahedron-sphere grid, one has to be careful in padding each patch with the contents of the other two neighboring patches. If a point is on the colored cut-lines as shown in Fig.4(c), then its \( k \)-ring hexagon neighbors are retrieved across the boundary of matrix. Notice here that by using the cross-boundary neighborhood padding strategy, the rectilinear data structure realizes a seamless geodesic grid representation of the 2-sphere.
Global rotation invariance  
Due to the alt-az rotation equivariance property of our spherical convolution operator, the global alt-az rotation of the input spherical image will be transformed into the same rotation of extracted spherical descriptors. A global pooling layer can be used right after the spherical convolution layer to achieve a global alt-az rotation invariance. Two options can be adopted here, global average pooling (GAP) or global max pooling (GMP). GAP layer sums all the feature values on the sphere and then averages them to obtain a single mean value to represent the whole sphere, whereas the GMP layer retains only the maximum feature point on the sphere. According to our experiments, both options perform similarly and both of them support alt-az global rotation invariance of the input shape. To make our method fully global rotation invariant in SO(3), a sub space augmentation with azimuthal rotation (rotate about the north pole) is desired.

5 Experiments

5.1 Spherical Descriptors of 3D shapes

As a pre-processing step for all experiments, we first need to convert 3D shapes to functions on the 2-sphere. Two different methods were employed to do this conversion: spherical projection for rigid shapes and spherical parameterization for non-rigid shapes.

Spherical projection  
For rigid shapes, we project a 3D shape onto an enclosing sphere using a straightforward ray casting scheme and we collect the following two types of spherical descriptors. 
(1) Spherical Extent Function (SEF): This function describes a surface by associating each ray from the origin to the distance of the last point of intersection of the model with the ray. (2) Normal Deviation Function (NDF): This function describes the surface using the cosine angle between surface normal at the last intersection point and the ray direction (see Fig. 5 left for example). Ignoring high non-convexity of surfaces, we assume the projections capture sufficient information of the shape to be useful for rigid shape analysis.

Spherical parametrization  
Spherical projection produces extrinsic shape descriptors which are not suitable for non-rigid shape analysis. To handle deformable shapes, we use the authalic spherical parametrization method (Sinha et al. (2016)) to obtain an area-preserving bijective spherical map and use the following intrinsic shape descriptors: (1) Principal Curvatures: the two principle curvature \( k_{\text{min}} \) and \( k_{\text{max}} \) measure the degree to which the surface bends in orthogonal directions at a point. (2) Average Geodesic Distance (AGD): this measures the centerness of a point on the surface. (3) Heat kernel signature (HKS) (Sun et al. (2009)): this measures the amount of untransferred heat after time \( t \), assuming an unit heat source is added on each point of the surface (see Fig. 5 right for example).

In all of the following experiments, spherical functions are discretized using icosahedron-sphere grid with subdivision frequency \( f = 32 \). This will generate five patches of size \( 33 \times 65 \), by stacking them one above the other. The input size to all networks are \( 165 \times 65 \times K \), where \( K \) is the number of input channels. for the 12 valence-5 vertices in the icosahedron, we apply the shared hexagon.
Table 1: SHREC’11 classification result using different rotation modes with respect to three types of shape descriptors. Here, NA = no augmentation, AZ = augmenting training data with azimuthal rotations, Alt-AZ = augmenting training data with alt-az field rotations and SO(3) = augmenting training data with random rotations.

<table>
<thead>
<tr>
<th></th>
<th>NA</th>
<th>AZ</th>
<th>Alt-AZ</th>
<th>SO(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic-2</td>
<td>47.2%</td>
<td>68.9%</td>
<td>99.7%</td>
<td>72.5%</td>
</tr>
<tr>
<td>Intrinsic-3</td>
<td>75.9%</td>
<td>91.6%</td>
<td>99.7%</td>
<td>92.6%</td>
</tr>
<tr>
<td>Intrinsic-8</td>
<td>94.4%</td>
<td>100%</td>
<td>100%</td>
<td>99.1%</td>
</tr>
</tbody>
</table>

Table 2: ModelNet classification result using different perturbation modes of the testing data, demonstrating the rotation invariance property of our network with different types of unseen orientations. Here, NR = non-rotated and X/Y denotes, that the network was trained on X and evaluated on Y.

<table>
<thead>
<tr>
<th></th>
<th>NR/NR</th>
<th>NR/AZ</th>
<th>NR/ALT-AZ</th>
<th>NR/SO(3)</th>
<th>SO(3) / SO(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelNet10</td>
<td>93.3%</td>
<td>84.0%</td>
<td>91.5%</td>
<td>90.2%</td>
<td>89.0 %</td>
</tr>
<tr>
<td>ModelNet40</td>
<td>89.6%</td>
<td>73.4%</td>
<td>89.4%</td>
<td>87.9%</td>
<td>88.7%</td>
</tr>
</tbody>
</table>

filter by computing the center point twice. Since it affects a small number of vertices, we empirically validate that the effect can be ignored.

5.2 Non-rigid shape classification

We first conduct experiments on SHREC’11 non-rigid shape classification, and we compare three types of spherical functions: (i) Intrinsic-2 contains the two principle curvatures, (ii) Intrinsic-3 adds AGD to intrinsic-2, and (iii) Intrinsic-8 adds five HKS sampled at 5 logarithmic time scales on top of Intrinsic-3. And we compare four modes of experiments: (a) trained with original data without data augmentation (NA), (b) trained with 36 azimuth rotation augmentation (AZ) by sampling \( \omega \) per 10 degrees, (c) trained with 72 alt-az rotation augmentation (Alt-AZ) by sampling \( \theta \) and \( \phi \) per 30 degrees, and (c) trained with arbitrary 128 rotations (SO(3)). In each category, 16 objects are used for training and 4 object are used for testing.

Architecture and hyperparameters Our network contains five \( a^3 \)SConv-dropout-ReLU-SPooling blocks (here, \( a^3 \)SConv represents an alt-az spherical convolution layer, and SPooling represents a spherical pooling layer). A 20% dropout layer is added right after each spherical convolution layer for regularization. The resulting spherical functions are pooled using global average pooling (GAP) layer followed by two fully connected layers for the final classification. A 50% dropout layer is inserted in between the last two fully connected layers. We use 32, 64, 64, 128, 128 features for the \( a^3 \)SConv layers, and 512 features in the first fully connected layer. Each filter on \( S^2 \) has kernel size ring-2, stride 1 and each SPooling layer has size ring-2 and stride 2.

Results Table 1 shows the performance of these intrinsic descriptors for non-rigid shape classification under deferent augmentation modes. Notice that the original testing data are randomly posed. Inspite the small training data, our network achieves good classification accuracy for Intrinsic-8 without data augmentation. For inputs with less channels, augmentation is necessary because the of the small size of training data. Notice pure local geometric information (Intrinsic-2) can only be used to classify the SHREC’11 dataset with enough data augmentation. Augmenting the inputs with rotations is beneficial due to interpolation and sampling effects. Comparing the three types of data augmentation strategies, in theory, only azimuthal type of augmentation is required, but according to the result, the alt-az and SO(3) augmentation performs better, we believe this is due to the increased number of training data that accounts for the discretization error. For deep learning based non-rigid shape analysis, the geometry image method (Sinha et al. (2016)) is most similar to ours. Their reported classification accuracy of 96.6% is based on a alt-az rotation augmentation. Our method outperforms the state-of-the-art approach by about 3% margin even by using two principle curvature (Intrinsic-2) as inputs.
5.3 RIGID SHAPE CLASSIFICATION

We further experiment on ModelNet10 and ModelNet40 rigid shape databases, and we use the model trained and tested on aligned data as baseline, and we explicitly test the equivariance/invariance property of our learned shape representations by perturbing the testing data in different modes.

**Architecture and experiment setup** We experiment with four types of perturbations: (a) test with original aligned data (NR), (b) test with azimuthal rotation perturbations (AZ), (c) test with alt-az field rotation perturbations (Alt-AZ) and (d) test with random $SO(3)$ rotations. We also randomly perturb the training data and test it with randomly perturbed testing data. In these experiments, two channels of spherical functions: SEF and NDF are used as the input and we use the same network structure as we use in SHREC’11, except that in the five cascaded $a^3$Sconv layers, 32,64,128,256,512 filters are used, and in the first fully connected layer, 1024 features are used.

**Results** Table 2 summarizes the performance of $a^3$SCNN for classifying rigid objects for unseen orientations. Without any rotation augmentation, nor any other sort of augmentation is used in these experiments. The column NR/NR shows the classification accuracy for aligned training and testing data. As expected, aligned data gives the best classification performance and it serves as the baseline for evaluating the rotation equivariance/invariance of the learned models. Comparing the four types of perturbing modes, the learned representation has the best classification accuracy for testing data perturbed with alt-az field rotation, this validates the alt-az rotation equivariance property of the learned model. For random $SO(3)$ rotation, whether it is with perturbed testing data only (see column NR/$SO(3)$) or with both testing and training data perturbed (column $SO(3)/SO(3)$), our network still performs well, showing that alt-az anisotropic 2-sphere convolution may generalize to unseen orientations. Among the three types of rotations, azimuthal type of perturbation gives slightly worse performance (see column NR/AZ). This is in line with our expectation because, alt-az field rotation group is a subgroup of $SO(3)$, but there is no intersection between azimuthal rotation and alt-az rotation. Even though our network achieves some level of azimuthal rotation equivariance, we believe this is partly contributed by the spherical pooling layer and global average pooling layer.

**Discussion** It is difficult to provide quantitative comparison because little research has been done on learning rotation invariant shape descriptors. The state-of-the-art methods such as, volumetric methods and point cloud based method report very high classification accuracies on ModelNet, but none of them can generalize to unseen orientations. Given the rather task agnostic architecture of our model and the lossy but compact input representation we use, we interpret our models performance as strong empirical support of the effectiveness of learning rotation invariant shape descriptors using alt-az spherical convolution operators.

5.4 3D SHAPE RETRIEVAL

We evaluate shape retrieval performance on the challenging SHREC’11 non-rigid dataset. *Intrinsic-d* is used as our input spherical images and we extract the output 512 features of the fully connected layer as the shape descriptors. Our approach significantly outperforms all other methods with 0.82 mAP retrieval performance. Fig[6] uses the dimensionality reduction method t-SNE \cite{van2008visualizing} to plot the rotation invariant feature descriptors extracted. It shows that our learned descriptors successfully disentangle the original 3D object space and exhibit a clustered behavior in the feature vector space.

Finally, we run shape retrieval experiments on ShapeNet Core55, following rules of the SHREC’17 3D shape contest \cite{Savva2016}. There is a aligned regular dataset and a version in which all models are perturbed by rotations. We concentrate on the perturbed version to test the quality of our learned alt-az rotation invariant shape descriptors for large scale rigid 3D object retrieval task.

The same architecture and same type and size of input that we used for ModelNet classification problem is used in this experiment. The learned model from ModelNet40 was tranfered and fine-tuned for SHREC’17 feature extraction. Our trained model obtaines a 83% classification accuracy using the validation subset and we use the 1024 features extracted from the first fully connected layer as the invariant shape descriptors to perform the shape similarity calculation using cosine distance.
Figure 6: The shape descriptor of SHREC’11 original models (training set) extracted by our $\alpha^3$SCNN network, rendered with t-SNE.

Table 3: Results and best competing methods for the SHREC17 competition, perturbed dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>P@N</th>
<th>R@N</th>
<th>F1@N</th>
<th>mAP</th>
<th>NDCG</th>
<th>Input size</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tatsuma &amp; Aono (2009)</td>
<td>0.705</td>
<td>0.769</td>
<td>0.719</td>
<td>0.696</td>
<td>0.783</td>
<td>38 × 224$^2$</td>
<td>3M</td>
</tr>
<tr>
<td>Furuya &amp; Ohbuchi (2016)</td>
<td>0.814</td>
<td>0.683</td>
<td>0.706</td>
<td>0.656</td>
<td>0.754</td>
<td>126 × 10$^3$</td>
<td>8.4M</td>
</tr>
<tr>
<td>Cohen et al. (2018)</td>
<td>0.701</td>
<td>0.711</td>
<td>0.699</td>
<td>0.676</td>
<td>0.765</td>
<td>6 × 128$^2$</td>
<td>1.4M</td>
</tr>
<tr>
<td>Esteves et al. (2018)</td>
<td>0.690</td>
<td>0.684</td>
<td>NA</td>
<td>0.630</td>
<td>NA</td>
<td>2 × 32$^2$</td>
<td>0.5M</td>
</tr>
<tr>
<td>Ours</td>
<td>0.701</td>
<td>0.702</td>
<td>0.695</td>
<td>0.650</td>
<td>0.762</td>
<td>2 × 165 × 65</td>
<td>1.8M</td>
</tr>
</tbody>
</table>

We then evaluated our trained descriptors using the official metrics and compared to the top four competitors, which includes the other two spherical convolution based methods. As shown in Table. 3, all three spherical convolution based methods (Cohen et al. (2018), Esteves et al. (2018) and ours) perform slightly below the current best, we believe this is due to the information loss caused by projecting 3D shapes onto the 2-sphere. To our surprise, all the three spherical convolution based methods report very similar performance, ours is slightly below Cohen et al. (2018) and slightly above Esteves et al. (2018). Both of the other two spherical convolution methods utilize Fast Fourier Transform (FFT) to compute the convolution which does not support local filters. Ours offers an alternative method which complete the current work while offers multi-level feature extraction capabilities.

6 Conclusion

In this paper, we presented and analyzed a convolutional neural network based on alt-az anisotropic spherical convolution operator which is different from the existing types of networks. Numerically, we implemented an efficient algorithm for computing spherical convolution with locally-supported geodesic filters using icosahedron-sphere grid. We demonstrated the efficacy of our approach for non-rigid/ rigid shape classification and retrieval and showed that it compares favorably to competing methods. Furthermore, we have shown that the proposed method can effectively generalize across rotations, and achieve state-of-the-art results on competitive 3D shape recognition tasks, without excessive data augmentation, feature engineering and task-tuning.
REFERENCES


