DEEP LEARNING 3D SHAPES USING ALT-AZ ANISOTROPIC 2-SPHERE CONVOLUTION

Min Liu[*] Purdue University
Fupin Yao Purdue University
Chiho Choi Honda Research Institute, USA.
Sinha Ayan Magic Leap Inc.
Karthik Ramani Purdue University

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 challenge 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 spherical patterns and so to extract geometric 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 (translational) 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 (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 require 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.

[*]Corresponding author. Email:liu66@purdue.edu

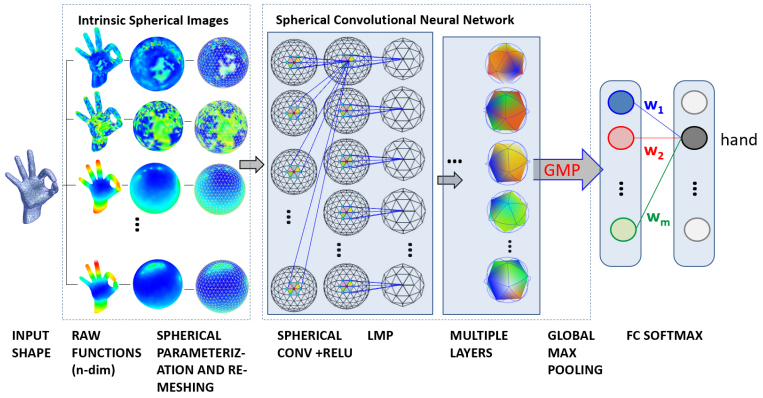

Figure 1: An example of our alt-az anisotropic spherical convolution neural network (a^3S-CNN) applied for a non-rigid shape classification problem.

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); Monti et al. (2017)) uses local geodesic polar coordinates to parameterize a surface patch locally around a point. An angular max-pooling layer or local coordinate frame alignment are 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 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)), ominidirectional 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 $\mathrm{SO}(3)$ 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 a^3SCNN 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 a^3SCNN 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 standard 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^3SCNN with 3D shape learning problems.

3 ALT-AZ CONVOLUTION ON 2-SPHERE

3.1 NOTATIONS AND PRELIMINARIES

2-sphere or unit sphere $\mathbb{S}^2$ 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 $\hat{\mathbf{u}}$ is a unit vector. The $\hat{\mathbf{u}}$ can be parametrized by spherical coordinates $(\theta, \phi) \in [0, \pi] \times [0, 2\pi]$ such that $\hat{\mathbf{u}}(\theta, \phi) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$. A regular region r on $\mathbb{S}^2$ has a positive area $A = \int_r ds(\hat{\mathbf{u}})$, where $ds(\hat{\mathbf{u}}) = \sin\theta d\theta d\phi$.

A special region on the 2-sphere is called polar cap region R_{θ_0}, around the north pole, $\hat{\eta}(0, 0, 1)$, which is azimuthally symmetric and is parameterized by a maximum colatitude angle θ_0:

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

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

$$\mathbf{R} \equiv \mathbf{R}^{(zyz)}_{\varphi\vartheta\omega} \triangleq \mathbf{R}^{(z)}_{\varphi}\mathbf{R}^{(y)}_{\vartheta}\mathbf{R}^{(z)}_{\omega} \tag{2}$$

If we fix the third rotation angle ω to zero, $\mathrm{SO}(3)$ is reduced into a subset $\mathfrak{A}$ with two independent parameters. Any rotation $\mathcal{R} \in \mathfrak{A}$ can be described as an **alt-az rotation** [1]:

$$\mathcal{R} \equiv \mathbf{R}^{(zyz)}_{\varphi\vartheta 0} \triangleq \mathbf{R}^{(z)}_{\varphi}\mathbf{R}^{(y)}_{\vartheta} \tag{3}$$

An alt-az rotation can be considered as a composition of an altitude rotation $\mathbf{R}^{(y)}_{\vartheta} \in \mathrm{SO}(2)$ and a azimuth rotation $\mathbf{R}^{(z)}_{\varphi} \in \mathrm{SO}(2)$.

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

$$(\mathcal{D}_R(\varphi, \vartheta, \omega)f)(\hat{\mathbf{u}}) = f(\mathbf{R}^{-1}\hat{\mathbf{u}}). \tag{4}$$

3.2 CONVOLUTION ON THE 2-SPHERE

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

$$(h \otimes f)(\mathbf{x}) \triangleq \int_{\mathbb{R}^n} h(\mathbf{x} - \mathbf{y})f(\mathbf{y})d\mathbf{y}, \qquad \mathbf{x} \in \mathbb{R}^n \tag{5}$$

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

[1] To avoid the ill definition at the two poles, when applied for spherical convolution, we constrain the alt-az rotation by imposing the following condition: if $\vartheta = 0$ or $\vartheta = \pi$, then $\varphi = 0$.

Unlike conventional Euclidean domain signal, for spherical functions there is no standard convolution operators defined. 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 $\mathrm{SO}(3)$ 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 $\hat{\mathbf{u}} \in \mathbb{S}^2$, general anisotropic convolution on $\mathbb{S}^2$ is defined:

$$h \boxdot f(\mathbf{R}) = g(\varphi, \vartheta, \omega) \triangleq \int_{\mathbb{S}^2} \sum_{k=1}^{K} (\mathcal{D}_R(\varphi, \vartheta, \omega)h)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) \tag{6}$$

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 $\mathrm{SO}(3)$ (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 $h_0(\hat{\mathbf{u}})$ (Esteves et al. (2018); Driscoll & Healy (1994)):

$$h_0(\hat{\mathbf{u}}) \triangleq \int_0^{\pi} \frac{1}{2\pi} (\mathcal{D}_{Rz}(\omega)h)(\hat{\mathbf{u}}) d\omega \tag{7}$$

$$h \circledcirc f(\mathbf{R}) = g(\varphi, \vartheta) \triangleq \int_{\mathbb{S}^2} \sum_{k=1}^{K} (\mathcal{D}_{Rz}(\varphi) \circ \mathcal{D}_{Ry}(\vartheta) h_0)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) \tag{8}$$

Referring to Eqn. (9), we see that an arbitrary filter h is essentially transformed into a rotationally 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 standard 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 $\mathrm{SO}(2)$ is generally not convolved. Several works on the rotation equivariant/invariant CNNs have been proposed (Weiler et al. (2018); Qiu et al. (2018); Kondor & Trivedi (2018),) and are proven to be effective; but they typically incur a significant increase in the number of parameters and computational load. Similarly, in the spherical domain, the two degrees of freedom in alt-az rotation are the direct analogs of two spatial translations in $\mathbb{R}^2$ ("shifting on the sphere"), and the third rotation $\mathbf{R}_{\omega}^{(z)}$, emulating the non-rotatable filters in $\mathbb{R}^2$, can be fixed and treated with data augmentation. Intuitively, we want to shift a spherical disc filter on the 2-sphere without self rotating the filter. We now formally define our alt-az spherical convolutional operator.

Type III: alt-az anisotropic spherical convolution (a^3SConv): Constraining the rotation of filter within alt-az rotation set $\mathfrak{A}$, a filter h spans the altitude change by ϑ and azimuth change by φ, and is convolved with the spherical signal f. Mathematically, a^3SConv is defined as:

$$(h \star f)(\mathbf{R}) = g(\varphi, \vartheta) \triangleq \int_{\mathbb{S}^2} \sum_{k=1}^{K} (\mathcal{D}_R(\varphi, \vartheta, 0)h)(\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}). \tag{9}$$

a^3SConv 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 extract hierarchical spherical patterns;
- **Azimuth rotation equivariance**: An map ℓ is rotation equivariant if $\ell \circ \mathcal{D}_Q = \mathcal{D}_Q \circ \ell$. In general cases, a^3SConv is not equivariant to an arbitrary rotation in $\mathrm{SO}(3)$. If Q is an azimuth rotation, a^3SConv has the equivariance property[2]. I.e. for an azimuth rotation $\mathcal{D}_Q(0, 0, \omega)$,

[2]With two poles as the singular points. Please see the proof in appendix A.

$$(h \star \mathcal{D}_Q f)(\mathbf{R}) = \mathcal{D}_Q (h \star f)(\mathbf{R}) \tag{10}$$

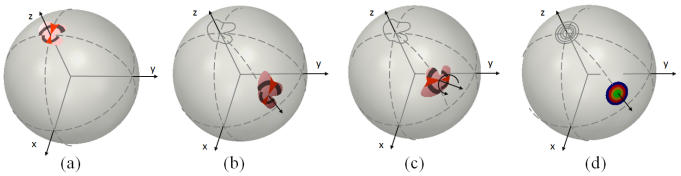

Figure 2: Rotation operators applied on a locally-supported kernel function. (a) An anisotropic kernel function h defined on a polar cap; (b) applying an alt-az rotation $\mathcal{D}_R(\pi/4, \pi/3, 0)$ to h; (c) applying a general rotation $\mathcal{D}_R(\pi/4, \pi/3, \pi/2)$ to h; (d) applying a general rotation $\mathcal{D}_R(\pi/4, \pi/3, \pi/2)$ 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. On the 2-sphere, we propose to use locally-supported geodesic disc filters in the form of polar caps. Mathematically, a locally-supported filter is defined as a space limited spherical function belonging to the follow subspace:

$$H_{R_{r0}}(\mathbb{S}^2) \triangleq \{h \in L^2(\mathbb{S}^2) : h(\theta, \phi) = 0, \forall \theta > r0\}, \tag{11}$$

where R_{r0} is the polar cap region on which the geodesic disc filter is defined, and $r0$ defines the size of a filter. Fig. 2 shows a locally-supported geodesic disc filter undergoing different types of rotation.

4 Spherical Convolutional Neural network

Our a^3SCNN consists of several layers that are applied subsequently (see Fig. 1). Besides the a^3SConv layer described above, we further discuss the following two specific types of layers defined on the 2-sphere.

A **local max pooling (LMP)** layer replaces a spherical image f^{in} at any point $\hat{\mathbf{u}}_0(\theta_0, \phi_0)$ with the maximum function value in its geodesic disc neighborhood, i.e.,

$$f^{out}(\hat{\mathbf{u}}_0) = \max_{|\hat{\mathbf{u}} - \hat{\mathbf{u}}_0| \leq r0} \{f^{in}(\hat{\mathbf{u}})\}, \tag{12}$$

where $\hat{\mathbf{u}}(\theta, \phi)$ is a neighboring point of $\hat{\mathbf{u}}_0$, and $|.|$ denotes the geodesic distance between them.

A **global spherical max pooling (GMP)** layer operates on a spherical image f^{in} with k channels and outputs a k dimensional vector in $\mathbb{R}^k$. For each channel f_i^{in} $(i = 1, 2, ..., k)$, a GMP layer outputs a single value represent the most salient feature. I.e.,

$$f_i^{out}(f^{in}) = \max_{\mathbb{S}^2} \{f_i^{in}\} \tag{13}$$

Notice a GMP layer is invariant to any rotation $\mathcal{D}_R(\varphi, \vartheta, \omega)$ of the input spherical image f: $\mathrm{GMP}(\mathcal{D}_R f) = \mathrm{GMP}(f)$.

Data augmentation After going through a set of a^3SConv layers, the global azimuth rotation of an input spherical image f will be transformed into the same rotation of the extracted spherical descriptors (see Eqn. (10)). With a GMP layer followed, the extract feature vector will be invariant to the azimuth rotation of f. For arbitrary rotation of f in $\mathrm{SO}(3)$, our a^3SConv layer does not have the equivariance property. This means data rotation augmentation is theoretically required to recognize f in random orientations. In appendix B, we show that, an a^3SCNN network constructed by several a^3SConv layers together with a GMP layer can generalize to arbitrary unseen orientations with $\mathrm{SO}(2)$ rotation augmentation about any axis which is not parallel to y or z axis.

5 NUMERICAL COMPUTATION OF ALT-AZ ANISOTROPIC CONVOLUTION

From a computational point of view, implementing the spherical convolution defined above in Eqns. (8-10) 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 point-wise 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 space limited 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.

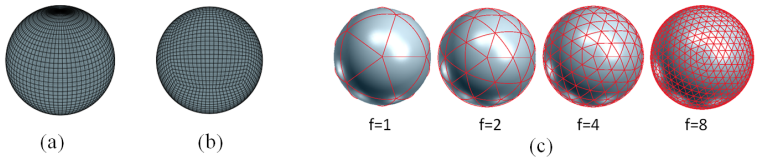

Figure 3: Different geodesic grids on the sphere. (a) a lat-lon grid, (b) a cubed-sphere grid and (c) icosahedron-sphere grid: from left to right,: one-frequency, two-frequency, four-frequency and eight-frequency subdivisions.

Geodesic grid discretization Uniform geodesic grid cannot be achieved on the sphere except for the projections of five platonic polyhedra - tetrahedron, cube, octahedron, dodecahedon 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 (Schröder & Sweldens (1995)). 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. 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. The same shape of geodesic disc and discretized hexagonal grid is used for local spherical max pooling LMP layers.

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.

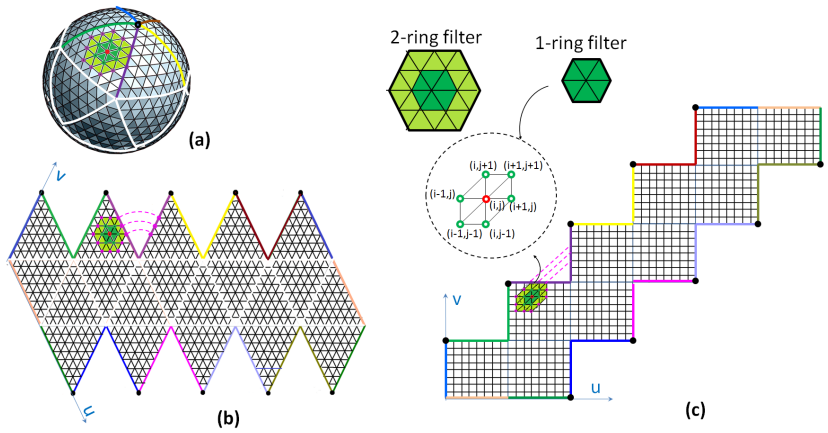

Figure 4: 2D rectilinear data structure for icosahedron-sphere grid. (a) The original spherical mesh of 8-frequency; (b) an icosahedron-sphere grid is opened onto a 2D plane, and the colored edges in the top and bottom indicate the locations along which the spherical grid is opened. (c) By rotating the two axes u and v, the flattened icosahedron in (b) can be stretched into a rectilinear grid structure represented by five 2D matrices.

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.

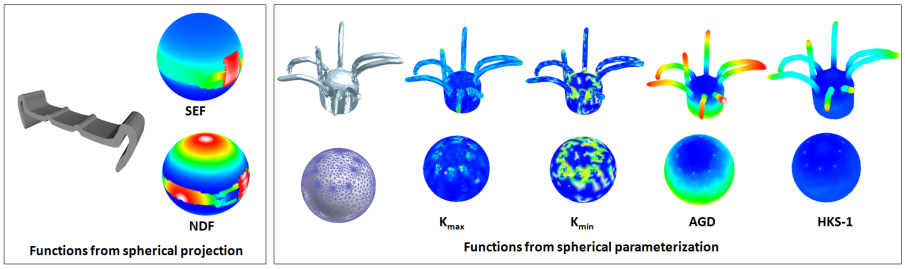

Figure 5: Spherical functions for non-rigid shapes using spherical projection method (left) and spherical parametrization and function mapping method (right). Here HKS-1 shows the first HKS channel.

6 Experiments

6.1 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

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 azimuth rotations about z-axis, SO(2)(x)=augmenting training data with SO(2) rotations about x-axis , Alt-AZ = augmenting training data with alt-az rotations and SO(3) = augmenting training data with random rotations.

	NA	AZ	SO(2)(x)	Alt-AZ	SO(3)
Intrinsic-2	47.2%	68.9%	89.6%	99.7%	72.5%
Intrinsic-3	75.9%	91.6%	94.4%	99.7%	92.6%
Intrinsic-8	94.4%	100%	100%	100%	99.1%

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 principal curvature k_{min} and k_{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 33×65, 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 filter by computing the center point twice. Since it affects a small number of vertices, we empirically validate that the effect can be ignored.

6.2 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 principal 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 five modes of experiments: (a) trained with original data without data augmentation (NA), (b) trained with 36 azimuth rotation augmentation (AZ) by sampling ω per 10 degrees, (c) trained with 36 rotation augmentation (SO(2)(x)) by rotating about x-axis per 10 degrees, (d) trained with 72 alt-az rotation augmentation (Alt-AZ) by sampling θ and ϕ per 30 degrees, and (e) trained with arbitrary 128 rotations (SO(3)). In each category, 16 objects are used for training and 4 objects are used for testing.

Architecture and hyper parameters Our network contains five a^3SConv-dropout-ReLU-LMP blocks. A 20% dropout is added right after each spherical convolution layer for regularization. The resulting spherical functions are pooled using a global max pooling (GMP) 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^3SConv layers, and 512 features are output from the GMP and fed into the first fully connected layer. Each filter on S^2 has kernel size ring-2, stride 1 and each LMP layer has size ring-2 and stride 2.

Results Table 1 shows the performance of these intrinsic descriptors for non-rigid shape classification under different augmentation modes. Notice that the original testing data are randomly posed. In spite the small training data, our network achieves good classification accuracy for *Intrinsic-8* even without data augmentation.We attribute the capability to generalize to random perturbed data to the use of LMP layers, which allows a certain amount SO(3) rotation invariance. Comparing the four types of data augmentation strategies, our experimental result confirms that SO(2)(x) aug-

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.

	NR/NR	NR/AZ	NR/ALT-AZ	NR/SO(3)	SO(3) / SO(3)
ModelNet10	93.3%	84.0%	91.5%	90.2%	89.0 %
ModelNet40	89.6%	73.4%	89.4%	87.9%	88.7%

mentation performs better than the original training data, AZ type augmentation and $\mathrm{SO}(3)$ random augmentation. It is predictable that alt-az augmentation performs even better with more augmented data. Theoretically, our network is invariant to azimuth rotation except for the two poles. Due to the approximation error introduced in the implementation, we see AZ type data augmentation can compensate those equivariance errors. Compare to other 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 also based on an alt-az rotation augmentation. Our method outperforms the state-of-the-art approach by about 3% margin even by using two principal curvature (*Intrinsic-2*) as inputs.

6.3 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 $\mathrm{SO}(3)$ 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^3Sconv layers, 32,64,128,256,512 filters are used, and in the first fully connected layer, 1024 features are generated for classification.

Results Table 2 summarizes the performance of a^3SCNN for classifying rigid objects for unseen orientations. 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 rotation. For random $\mathrm{SO}(3)$ rotation, whether it is with perturbed testing data only (see column NR/ $\mathrm{SO}(3)$) or with both testing and training data perturbed (column $\mathrm{SO}(3)$ / $\mathrm{SO}(3)$), our network still performs well, showing that a^3SCNN may generalize to unseen orientations even without data augmentation. It is counter-intuitive that azimuthal type of perturbation gives slightly worse performance (see column NR/AZ) since in theory, our network is azimuthal rotation invariant. We interpret this as the result of the equivariance error coming from icosahedron-sphere grid tessellation and singularities at poles, while alt-az and $\mathrm{SO}(3)$ perturbation might compensate this with perturbations which can be better treated by the LMP and GMP layers.

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 alt-az rotation invariant shape descriptors using alt-az spherical convolution operators.

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

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

6.4 3D SHAPE RETRIEVAL

We evaluate shape retrieval performance on the challenging SHREC'11 non-rigid dataset. *Intrinsic-8* 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 (van der Maaten & Hinton (2008)) 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.

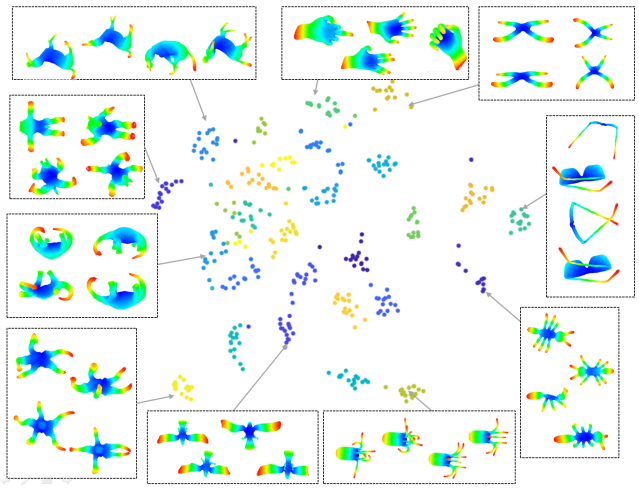

Figure 6: The shape descriptor of SHREC'11 original models (training set) extracted by our a^3SCNN network, rendered with t-SNE.

Finally, we run shape retrieval experiments on ShapeNet Core55, following rules of the SHREC'17 3D shape contest (Savva et al. (2016).) 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 shape descriptors to unseen orientations for large scale rigid 3D object retrieval task. An $\mathrm{SO}(2)$ rotation augmentation is performed by rotating each training data per 60 degrees about a random axis.

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 transferred and fine-tuned for SHREC'17 feature extraction. Our trained model obtains a 83% classification accuracy after about 24 hours of training 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. 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) preform slightly below the current best, we believe that 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 and GPU based fast computation.

7 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

S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki. Gift: A real-time and scalable 3d shape search engine. In *CVPR*, pp. 5023–5032, 2016.

W. Boomsma and J. Frellsen. Spherical convolutions and their application in molecular modelling. In *Advances in Neural Information Processing Systems (NIPS) 30*, pp. 3433–3443, 2017.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

Z. Cao, Q. Huang, and K. Ramani. 3d object classification via spherical projections. In *Proceedings of 3DV*, 2017.

T. S. Cohen, M. Geiger, J. Khler, and M. Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

J. R. Driscoll and D. M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Adv. Appl. Math.*, 15(2):202–250, June 1994.

C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so(3) equivariant representations with spherical cnns. In *European Conference on Computer Vision, ECCV 2018 (oral)*, 2018.

A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Computer Vision - ECCV 2004*, pp. 224–237, 2004.

T. Furuya and R. Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 121.1–121.12, September 2016.

X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(8): 949–958, 2004.

M. Kazhdan, T.s Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 156–164, 2003.

R. Kondor and S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, pp. 2747–2755, 2018.

A. Makadia and K. Daniilidis. Spherical correlation of visual representations for 3d model retrieval. *Int. J. Comput. Vision*, 89(2-3):193–210, 2010. ISSN 0920-5691.

H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.*, 36(4): 71:1–71:10, 2017. ISSN 0730-0301.

J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *ICCV Workshops*, pp. 832–840, 2015.

D. Maturana and S. Scherer. Voxnet:a 3d convolutional neural network for real-time object recognition. In *IROS*, pp. 922–928, 2015.

F. Monti, D. Boscaini, J. Masci, E. Rodol, and J. Svoboda and. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pp. 5425–5434, 2017.

C. Peng and S. Timalsena. Fast mapping and morphing for genus-zero meshes with cross spherical parameterization. *Computers & Graphics*, 59:107–118, 2016.

E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics (TOG)*, 22(3):340–349, 2003.

C. R. Qi, H. Su, M. Niener, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, pp. 5648–5656, 2016.

Q. Qiu, X. Cheng, R. Calderbank, and G. Sapiro. Dcfnet: Deep neural network with decomposed convolutional filters. In *ICML*, pp. 4198–4207, 2018.

G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, pp. 6620–6629, 2016.

M. Savva, F. Yu, Hao Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, Hang Su, S. Bai, X. Bai, N. Fish, J. Han, E. Kalogerakis, E. G. Learned-Miller, Y. Li, M. Liao, S. Maji, A. Tatsuma, Y. Wang, N. Zhang, and Z. Zhou. Large-scale 3d shape retrieval from shapenet core55. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, 3DOR '16, pp. 89–98, 2016.

P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *SIGGRAPH '95*, pp. 161–172, 1995.

K. Sfikas, I. Pratikakis, and T. Theoharis. Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218, 2018.

L. Shen and F. Makedon. Spherical mapping for processing of 3d closed surfaces. *Image and vision computing*, 24(7):743–761, 2006.

B. Shi, S. Bai, Z. Zhou, and X. Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In *ECCV*, pp. 223–240, 2016.

A. Sinha, A. Unmesh, Q. Huang, and K. Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *CVPR*, pp. 6040–6049, 2017.

S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, pp. 808–816, 2016.

H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pp. 945–953, 2015.

Y. Su and K. Grauman. Learning spherical convolution for fast features from 360 image. In *Advances in Neural Information Processing Systems (NIPS) 30*, 2017.

J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum*, 28(5):1383–1392, 2009.

A. Tatsuma and M. Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval. *Vis. Comput.*, 25(8):785–804, 2009.

L.J.P van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9: 25792605, 2008.

P. Wang, Y. Liu nd Y. Guo, C. Sun, and T. Xin. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72:1–72:11, 2017.

M. Weiler, F. A. Hamprecht, and M. Storath. Learning steerable filters for rotation equivariant cnns. In *CVPR*, pp. 849–858, 2018.

Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pp. 1912–1920, 2015.

Li Yi, Hao Su, Xingwen Guo, and Leonidas J. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *CoRR*, abs/1612.00606, 2016.

APPENDIX

A AZIMUTH ROTATION EQUIVARIANCE

Under the definition of alt-azimuth anisotropic convolution, for an alt-az rotation $D_{\mathbf{R}}(\varphi, \vartheta, 0)$, and a general rotation $D_Q(\varphi_1, \vartheta_1, \omega_1)$, (assume the number of channels $K = 1$ for simplicity,) we have:

$$\begin{aligned}(h \star \mathcal{D}_Q f)(\mathbf{R}) &= \int_{\mathbb{S}^2} (\mathcal{D}_R h)(\hat{\mathbf{u}}) f(\mathbf{Q}^{-1}\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) = \int_{\mathbb{S}^2} h(\mathbf{R}^{-1}\hat{\mathbf{u}}) f(\mathbf{Q}^{-1}\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) \\ &= \int_{\mathbb{S}^2} h(\mathbf{R}^{-1}\mathbf{Q}\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) = \int_{\mathbb{S}^2} h((\mathbf{Q}^{-1}\mathbf{R})^{-1}\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}})\end{aligned} \tag{14}$$

$\mathbf{Q}^{-1}\mathbf{R}$ is in general a rotation in $\mathrm{SO}(3)$, but when $\mathbf{Q}$ is an azimuth rotation, $\mathbf{Q}^{-1}\mathbf{R} = \mathbf{R}^{(z)}_{-\varphi_1}\mathbf{R}^{(z)}_{\varphi}\mathbf{R}^{(y)}_{\vartheta} = \mathbf{R}^{(z)}_{\varphi-\varphi 1}\mathbf{R}^{(y)}_{\vartheta}$ is an alt-az rotation such that,

$$(h \star \mathcal{D}_Q f)(\mathbf{R}) = \int_{\mathbb{S}^2} h((\mathbf{Q}^{-1}\mathbf{R})^{-1}\hat{\mathbf{u}}) f(\hat{\mathbf{u}}) ds(\hat{\mathbf{u}}) = (h \star f)(Q^{-1}\mathbf{R}) = \mathcal{D}_Q(h \star f)(\mathbf{R}) \tag{15}$$

B $\mathrm{SO}(2)$ ROTATION AUGMENTATION FOR GENERAL ORIENTATIONS

Since composite a^3SConv layers will not affect rotation equivariance property, without losing generality, we assume our network consists of one a^3SConv and one GMP layer. For any feature output from a GMP layer, suppose it is activated at point $\hat{u_0}(\theta_0, \phi_0)$. It corresponds to a maximum correlation between the learned filter h and the input image f (when h is "alt-az" rotated onto $\hat{u_o}$). Any $\mathrm{SO}(3)$ rotation of f which first rotates the point $\hat{u_o}$ back to the north pole $\hat{\eta}(0, 0, 1)$, followed by an arbitrary alt-az rotation $D(\phi', \theta', 0)$ to new point $\hat{u_1}$, will be invariant to the network. When h is convolved at point $\hat{u_1}$. We define this set of rotation as "alt-az shift rotation"(see Fig.7(a)). Alt-az shift rotation rotation causes no relative angular change of the geodesic disc centered at $\hat{u_o}$ with respect to the h's convolving.

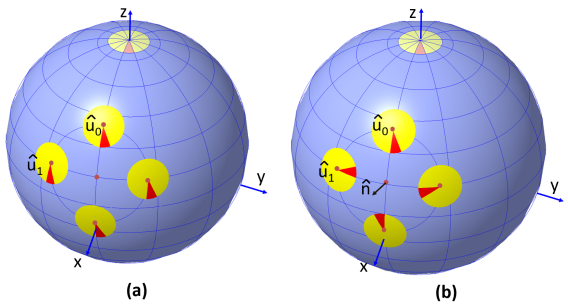

Figure 7: Rotations of salient features. (a) Alt-az shift rotations are invariant to the network, (b) a rotation of a salient feature about arbitrary axis. When local filter h is convolved at each rotated $\hat{u_1}$, there is a relative angular change compared with (a).

For a direction $\hat{n}$ ($\hat{n}$ is not along z-axis or y-axis), if one rotates f about $\hat{n}$ an arbitrary angle which move the original salient point $\hat{u_o}$ to $\hat{u_1}$, this rotation is in general not an alt-az shift rotation and will change the correlation between h and f when h is convolved at $\hat{u_1}$ (See Fig.7(b)). If one rotates f about $\hat{n}$ a full round, the geodesic disc centered at $\hat{u_o}$, will go through a relative self rotation from 0 to 360 degrees, using the alt-az rotation of h to each corresponding point as the direction references.

Therefore, by augmenting f using $\mathrm{SO}(2)$ rotation about an arbitrary axis $\hat{n}$, any feature of f, after $\mathrm{SO}(3)$ random rotation, can always alt-az shift rotate to the same feature in one of f's augmented copies.