IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 27, 2022, accepted February 7, 2022, date of publication February 14, 2022, date of current version March 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151350

Robust Object Classification Approach Using Spherical Harmonics

AYMAN MUKHAIMAR[1], (Member, IEEE), RUWAN TENNAKOON[2], CHOW YIN LAI[3], REZA HOSEINNEZHAD[1], AND ALIREZA BAB-HADIASHAR[1]

[1]School of Engineering, RMIT University, Melbourne, VIC 3000, Australia
[2]School of Science, RMIT University, Melbourne, VIC 3000, Australia
[3]Department of Electronic and Electrical Engineering, University College London, London WC1E 6BT, U.K.

Corresponding author: Alireza Bab-Hadiashar (alireza.bab-hadiashar@rmit.edu.au)

ABSTRACT Point clouds produced by either 3D scanners or multi-view images are often imperfect and contain noise or outliers. This paper presents an end-to-end robust spherical harmonics approach to classifying 3D objects. The proposed framework first uses the voxel grid of concentric spheres to learn features over the unit ball. We then limit the spherical harmonics order level to suppress the effect of noise and outliers. In addition, the entire classification operation is performed in the Fourier domain. As a result, our proposed model learned features that are less sensitive to data perturbations and corruptions. We tested our proposed model against several types of data perturbations and corruptions, such as noise and outliers. Our results show that the proposed model has fewer parameters, competes with state-of-art networks in terms of robustness to data inaccuracies, and is faster than other robust methods. Our implementation code is also publicly available at https://github.com/AymanMukh/R-SCNN

INDEX TERMS Object recognition, point cloud classification, spherical harmonics, robust classification.

I. INTRODUCTION

Objects detection and classification is a crucial part of many robotic manipulation applications [1]. For example, autonomous cars or robots can better interact with the surrounding environment if they accurately recognize objects. Due to the existence of compact, low-cost 3D scanners such as Microsoft Kinect and Intel RealSense, 3D object measurements are readily available. These scanners generate point clouds either using Light Detection and Ranging (LIDAR) or using stereo matching. The generated point clouds of these scanners are often noisy and contain outliers, which significantly deteriorates the accuracy of existing object classification methods.

The recent success and popularity of Convolutions Neural Networks (CNN) for many computer vision applications have inspired researchers to use them for 3D model classification as well [2]–[4]. To exploit the potential of deep networks for this application, different representations of 3D data have been proposed, including kd-tree [5], dynamic graphs [6], Random Sample Consensus (RANSAC) [7], [8], and most recently, spherical harmonics [9]–[12]. Spherical harmonics is a representation that have attracted significant interest in a wide range of applications including matching and retrieval [13], [14], lighting [15], and surface completion [16]. They attain several favourable characteristics for working with 3D space, such as their basis are defined on the surface of the sphere (volumetric) and are rotation equivariant. In addition, they have shown to provide compact shape descriptors compared to other types of descriptors [13], [17].

The use of CNNs with spherical harmonics has had major success in several recent papers for shape classification [9], [10], [12], retrieval [10] and alignment [10]. Unlike conventional approaches that use CNNs in regular Euclidean domains, spherical harmonics CNNs (SCNNs) apply convolutions in SO(3) Fourier space, learning features that are SO(3)–equivariant. Frameworks that use spherical harmonics CNNs can be divided into two groups: Point-based SCNN that extract features based on point maps or pairwise relations [11], [12] and the other group that uses spherical harmonics convolution on images casted on the sphere [9], [10]. Interestingly, spherical CNNs have shown to have fewer parameters [9] and faster training due to the reduction in the dimensionality of the spherical harmonics shape descriptors, which make them a suitable candidate for low-cost robots with limited computational power.

The associate editor coordinating the review of this manuscript and approving it for publication was Haiyong Zheng.

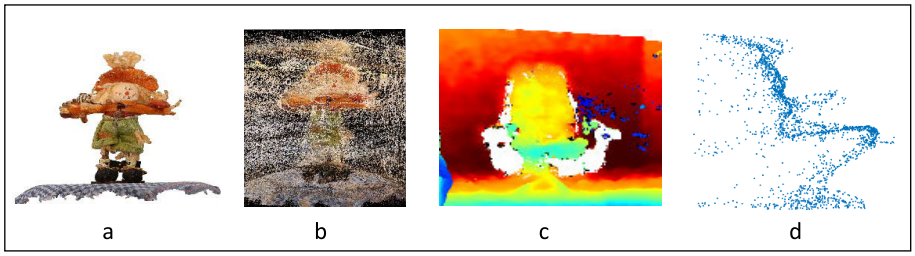

FIGURE 1. a: A 3D model of a scarecrow, b: the point cloud of the scarecrow generated using a multiview image pipline, data were taken from [18]. c: Point cloud of a scene captured by Intel RealSense scanner, and d: side of view of the chair captured in c.

Point clouds produced by either 3D scanners or multi-view images are often imperfect and contain noise and/or outliers. Several factors lead to those measurements inaccuracies in the point clouds such as adverse weather conditions, e.g. fog [19]–[21] and rain [22], [23], objects reflective surface, the scanner itself [24]–[26], or by some pipelines that construct 3D objects from multi-view images [18], [27], [28]. Examples of those data inaccuracies can be seen in Figure 1. The first image shows a scarecrow that was used for image-based 3D reconstruction. The second image shows the generated point cloud of the scarecrow using a 3D reconstruction algorithm [18], [29]. As seen in Figure 1-b, the generated point cloud has a high number of outliers surrounding the object. The third image shows a scene containing a chair that was captured by Intel RealSense laser 3D scanner. The surface of the chair is noisy and surround by outliers, as seen in the last image. These data inaccuracies make point cloud classification challenging. As such, the development of robust classification frameworks that can deal with such inaccuracies is needed for autonomous systems and robot object interactions.

This paper presents a spherical harmonics approach that is robust to the uncertainty in point clouds data. The proposed approach is computationally efficient as it requires no pre-processing or filtering of outliers and noise. Instead, our entire robust classification operation is performed in an end-to-end manner. To present our approach, we first discuss the spherical harmonics descriptors and the common sampling strategies used in the literature. We then show that using concentric spheres with density occupancy grids provides the highest robustness against data inaccuracies. We also propose using the magnitude of each specific spherical component for shape classification, and we show that it produces better robustness than using the combined magnitudes of different components at each order. In particular, we show that a simple classifier (i.e. fully connected neural network) with the previously mentioned spherical harmonics descriptors and sampling strategy is robust to high levels of data inaccuracies. Using the above knowledge and the inspiration from the recent success of spherical CNNs approaches [9], [10], we propose a light spherical convolutional neural network framework (called RSCNN) that is able to deal with different types of uncertainty inherent in three-dimensional data measurement.

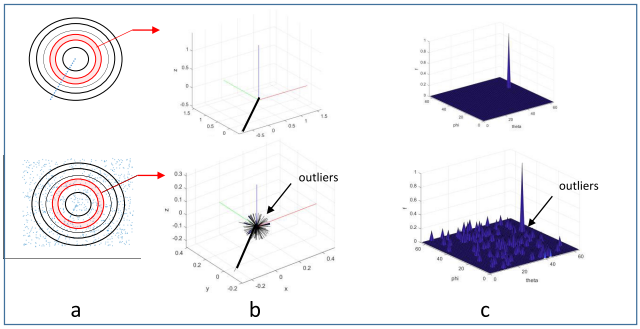

FIGURE 2. a: Sampling of a line using concentric spheres. The same line is corrupted with outliers in the second row. b: The third concentric sphere corresponding spherical function $f(\varphi, \theta)$. c: The same spherical function $f(\varphi, \theta)$ that is shown in b is plotted in the spherical coordinates.

Unlike previous approaches, our proposed framework performs the classification in the Fourier domain, where it's easier to determine similarities between noisy 3D objects. Moreover, unlike previous approaches [2], [4], [30], the applied spherical convolution operation is simply multiplying the filter kernels by the spherical harmonic coefficients, hence, the convolution operation does not disrupt the input signal through the use of a pooling operation or grid altering. We show that the output features produced by the convolution operation are highly robust.

Our experiments show that the use of concentric spheres with density occupancy grids provides high robustness to outliers and other types of data inaccuracies. To demonstrate the robustness of the proposed sampling along with the use of spherical harmonic transform, we present a simple case in Figure 2 of a line plotted in 3D. The line was corrupted with outliers as shown in the second row of the figure in a. Figure 2-b shows the spherical function $f(\varphi, \theta)$ of the third concentric sphere, where the distance from the origin (f) represents the number of points corresponding to each theta and phi (plots are in Cartesian for visualization). We plot the same figure in spherical coordinates in Figure 2-c. The maximum value of f is one as we are using a density occupancy grid (we divide by max). With the use of density occupancy grid, outliers appear as small peaks, while inliers have higher peaks, as can be seen in Figure 2-c. This representation makes outliers appear as small noise, where noise in Fourier transform (spherical harmonic transform) appears at high frequencies, and our experiments show that by using low frequencies, we avoid storing noise in our shape descriptors.

Our key contributions in this paper are as follows:

- We propose a CNN framework that is significantly more robust than existing approaches to point cloud inaccuracies generated by commercial scanners.
- The proposed approach requires no pre-processing steps to filter outliers or noise, but instead, the entire robust classification operation is performed in an end-to-end manner.
- The proposed approach uses compact shape descriptors (spherical harmonics) that reduce the size of our model, making it is suitable for low-cost robots with limited computational power. We demonstrate the efficiency and accuracy of our method on shape classification with the presence of several types of data inaccuracies, and we show that our framework outperforms all previous approaches.

II. METHODOLOGY

A. PRELIMINARIES

In this section, we review the theory of spherical harmonics along with their associated descriptors that are used for classification tasks. In addition, we review the theory of convolution operations applied to spherical harmonics.

1) SPHERICAL HARMONICS

Spherical harmonics are a complete set of orthonormal basis functions, similar to the sines and cosines in the Fourier series, that are defined on the surface of unit sphere $S2$ as:

$$Y_l^m(\theta,\varphi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{im\varphi} \tag{1}$$

where $P_{lm}(x)$ is the associated Legendre polynomial, l is the degree of the frequency and m is the order of every frequency ($l \geq 0, |m| \leq l$). $\theta \in [0,\pi]$, $\varphi \in [0,2\pi]$ denote the latitude and longitude, respectively. Any spherical function $f(\theta,\varphi)$ defined on unit sphere $S2$ can be estimated by the linear combination of these basis functions:

$$f(\theta,\varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \hat{f}_{lm} Y_l^m(\theta,\varphi) \tag{2}$$

where $\hat{f}_{lm}$ denotes the Fourier coefficient found from:

$$\hat{f}(l,m) = \int_0^{\pi} \int_0^{2\pi} f(\theta,\varphi) \bar{Y}_l{}^m(\theta,\varphi) \sin\varphi d\varphi d\theta \tag{3}$$

where $\bar{Y}$ denotes the complex conjugate. The following descriptors implies that any spherical function can be described in terms of the amount of energy $|\hat{f}|$ it contains at every frequency:

$$D1 = (|\hat{f}_{0,0}|, |\hat{f}_{1,0}|, |\hat{f}_{1,1}|, \ldots |\hat{f}_{lm}|), \tag{4}$$

or the amount of energy it contains at every degree:

$$D2 = (|\hat{f}_0|, |\hat{f}_1|, |\hat{f}_2|, \ldots |\hat{f}_l|), \ \textit{where}\ |\hat{f}_i| = \sqrt{\sum_{m=-i}^{i} \hat{f}_{im}^2} \tag{5}$$

Both of these descriptor vectors have been used for classification of shapes [13], [14]. However, only the second descriptor is rotation equavareint while the first one carries more shape information. We have investigated the use of both descriptors for shape classification and the result is provided in section III.

2) SPHERICAL CONVOLUTION

If we have a function f with its Fourier coefficients $\hat{f}$ found from Equation (3), and another function or kernel h with its Fourier coefficients $\hat{h}$, then the convolution operation in spherical harmonic domain is equal to the multiplication of both functions Fourier coefficients as shown below [31]:

$$(f * h)_l^m = \sqrt{\frac{4\pi}{2l+1}} \hat{f}_l^m \hat{h}_l^0. \tag{6}$$

Here, the convolution at degree l and order m is obtained by multiplying of the coefficient $\hat{f}_l^m$ with the zonal filter kernel $\hat{h}_l^0$. The inverse transform is also achieved by summing overall l values:

$$(f * h)(\theta,\varphi) = \sum_{l=0}^{\infty} \sum_{|m|\leq l} (f * h)_l^m Y_l^m(\theta,\varphi). \tag{7}$$

B. RELATED WORK

Spherical harmonics have been used for 3D shape classification for many years [13], [14]. Early classification frameworks used spherical harmonic coefficients as shape descriptors [13]. Later, with the use of spherical convolutions, classifier networks were empowered to learn descriptive features of objects. We study both approaches in terms of their performance under data inaccuracies.

The spherical CNNs proposed in [9], [10], for spherical signals defined on the surface of a sphere, addresses the rotation equavarience using convolutions on the set of the 3D rotation group (SO3) and S2 rotation group. In [9], the spherical input signal is convolved with S2 convolution to produce feature maps on SO3, followed by an SO3 convolution. While zonal filters are used in the spherical convolutions in [10]. Steerable filters [32]–[34] were used to achieve rotation equivariance. The filters use translational weight sharing over filter orientations. The sharing led to a better generalization of image translations and rotations. The network filters were restricted to the form of complex circular harmonics [32], or complex valued steerable kernels [33]. Sphnet [12] is designed to apply spherical convolution on volumetric functions [35] generated using extension operators applied on point cloud data. Unlike previous approaches, spherical convolution is applied on point clouds instead of a spherical voxel grid, resulting in a better rotation equivariant. In Deepsphere [11], spherical CNNs are used on graph represented shapes. The shapes are projected onto the sphere using HEALPix sampling, in which the relations between the pixels of the sphere build the graph. The graph is then represented by the Laplacian equation, which is solved using spherical CNNs. Ramasinghe *et al.* [36] investigated the use

of radial component in spherical convolutions instead of using spherical convolutions on the sphere surface. They proposed a volumetric convolution operation that was derived from Zernike polynomials. Their results show that the use of volumetric convolution provides better performance by capturing depth features inside the unit ball. Spherical signals have also been used in conjunction with conventional CNNs by [37]–[39] to achieve better rotational equivariance than signals on euclidean space. You *et al.* [40] used concentricity to sample 3D models with a sampling strategy that has better robustness to rotations. While previous approaches used spherical harmonics to build rotation equivariant neural networks, our focus is on building a robust spherical harmonics structure. As such, our choice of representation is not restricted to spherical harmonics descriptors that are rotation equivariant.

In terms of recent deep learning approaches for 3D shape classification, 3D CNNs have been used with voxel-based 3D models [3], [41]–[43] using several occupancy grids [41]. Such a representation has shown to be robust to data inaccuracies [43] while some implementations (e.g. [44]) have achieved very high classification accuracy for clean objects (using ModelNet40). Another approach is to use 2D CNN on images of the 3D mesh/CAD objects rendered from different orientations [4], [45], [46]. The rendered images are usually fed into separate 2D CNN layers; a pooling layer follows these layers to aggregate their information. These methods take advantage of existing pertained models to achieve high classification accuracy. When testing MVCNN [4], the classification accuracy was heavily affected by data inaccuracies, especially outliers. Another approach uses unsorted and unprocessed point clouds directly as an input to the network layers [2], [6], [30], [47]. These approaches use a max-pooling layer that was tested to be robust to point dropout and noise [48]. However, when tested with outliers, their performance was significantly affected. Another approach is to build upon relations between points [5]; for such methods, the existence of outliers completely changes the distance graph and causes such an approach to fail.

In terms of robust classification frameworks that exist in the literature, Pl-net3D [7] decompose shapes into planar segments and classify objects based on the segments information. DDN [49] proposes an end-to-end learnable layer that enables optimization techniques to be implemented in conventional deep learning frameworks. An m-estimators based robust pooling was proposed instead of max pooling used in conventional CNNs. Our approach shows better robustness to data argumentation while it involves less computation.

Although we focus in this paper on single object classification approaches, some applications require instance segmentation methods such as [2], [30], [50], [51]. Nevertheless, it is possible to achieve scene segmentation with single object classification methods using sliding box-based techniques [52], segmentation methods such as [53], [54], or utilizing a region proposal network [3].

III. OUR PROPOSED APPROACH

Unlike previous spherical harmonics approaches, our proposed framework, shown in Figure 3, combines the following three key contributions: the use of concentric spheres, the direct use of the magnitudes of spherical harmonics coefficients to classify objects, and limiting the spherical harmonics order level to mitigate the effect of noise/outliers to achieve robust classification. To describe our approach, we first introduce the problem, then we go through each step of the proposed solution and explain the overall framework.

A. PROBLEM STATEMENT

Point clouds of 3D models produced by either 3D scanners or 3D reconstruction algorithms are often imperfect and contain outliers. To test the robustness of previous methods on such scenarios, we recorded 56 scenes, by the Intel RealSense scanner, containing the following common objects: chairs, tables, cabinets, sofas, and desks. The recorded objects categories were chosen from the ScanObjectNN [55] dataset so that we can train methods on the ScanObjectNN training data, while we test them on our captured data. The ScanObjectNN contains real-scene 2.5D objects similar to our scans. However, our recorded objects have higher levels of noise due to the used scanner type and scanning range. Our recorded objects have noise and outliers with different levels. Examples of those objects can be seen in Figure 4. Moreover, to study the effect of each type of inaccuracies on machine learning models, we build comprehensive datasets by artificially adding noise, outliers, missing points, or a mixture of two types. We simulated those data inaccuracies, similar to previous studies [2], [6], [7], [30], [56], by adding uniformly distributed points in the object area (outliers), adding Gaussian noise to the point clouds, or randomly eliminating points from objects point clouds (missing points). We used Gaussian noise as we found out that distributions of measured noise of 3D scanners are somewhat Gaussian-like as shown in Figure 4-e. The standard deviation of the recorded noise was more than 0.1 in some cases. We measured noise by scanning a flat surface, fitting a plane to the point cloud of the surface, and finally recording the points to plane distance distribution. Besides using uniformly distributed outliers, we also test the robustness of recent methods on clustered outliers. Random point dropout simulates a case where the scanning density varies, i.e., objects are further from the scanner or using a scanner with lower scanning density. Examples of those data perturbations and corruptions can be seen in Figure 5.

B. SAMPLING ON CONCENTRIC SPHERES

The first step in using the spherical harmonics for modeling an object is to sample the input signal, which is referred to as $f(\varphi, \theta)$ in Equation (3). Two types of sampling are used in literature [13]: Sampling over the concentric spheres, and sampling over the sphere surface (image casting). For the first case, we generate a spherical voxel grid that consists of c concentric spheres with $n \times n$ grid resolution for each

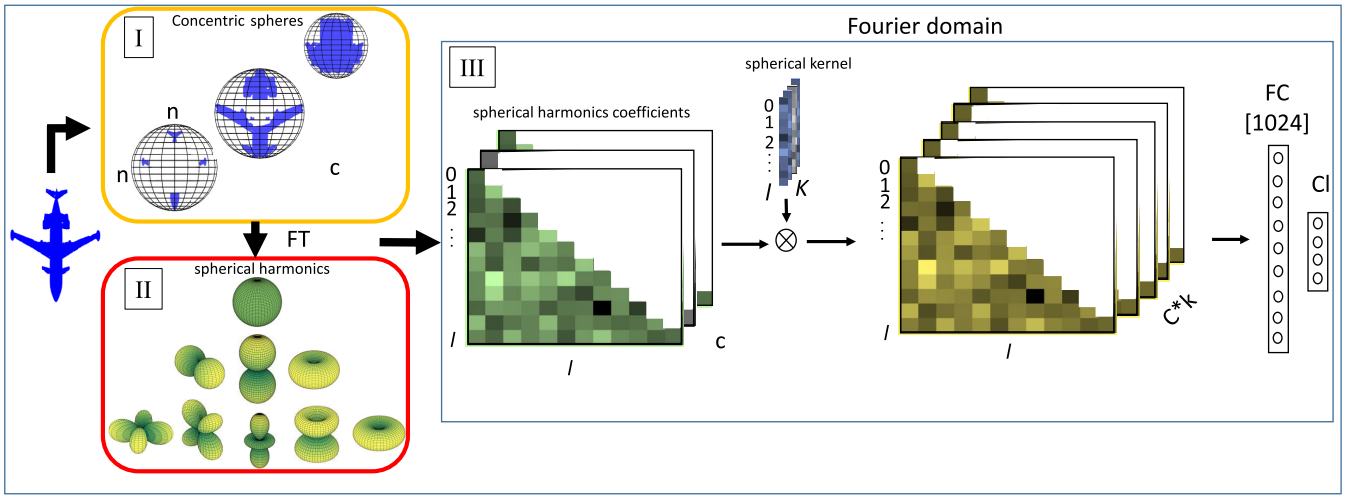

FIGURE 3. The proposed spherical CNN framework: I: We sample the shape with *c* concentric spheres and $n \times n$ grid resolution (section III-B). II: We apply Fourier transform (FT) on the spherical signal (section II-A1 and section III-C). We get the basis coefficients up to degree *l* for every concentric sphere (the first graph in section III of the figure). III: We apply the spherical convolution operation (the second graph in section III of the figure) on the basis coefficients, where k is the number of filters (section III-D). We then feed the spherical convolution output to a fully connected layer (FC) and a classification layer (Cl) (section III-E).

concentric sphere. The generated spherical voxel grid allows sampling over the unit ball ($S3$), with each voxel being represented by (r, θ, φ) where $(r = \{i-1, i\}, i \rightarrow 0 : c. \theta, \varphi = \{j-1, j\}, j * 2\pi/n, n \rightarrow 1 : n)$. We distribute the given 3D shape over the grid, and we keep a record of the number of points inside each voxel and produce a density occupancy grid. The use of such occupancy grid is expected to provide reliable estimates in the presence of outliers. We compare occupancy grids in the next section. To show the effect of noise and outliers on both sampling strategies, we considered a case study shown in Figures 6 and 7. Figure 6 shows sampling over sphere surface for a chair shown in column (a), with its corresponding spherical function $f(\varphi, \theta)$ shown in (b), while the generated shape after applying inverse transform (Equation 2) is shown in (c), and the reconstruction error between b and c is shown in (d). The first row shows the original shape, while the second row shows the shape corrupted with outliers, and finally, the third row shows the shape perturbed with noise. We used a degree number of $l = 10$ in those figures. As can be seen from the second row, outliers heavily affected the generated function $f(\varphi, \theta)$ shown in b, which affected the reconstructed shape in c. In contrast, the noise didn't substantially affect the constructed object when comparing images in c for the first and third rows. Noise in Fourier transform appears at high frequencies [57], while using low frequencies, such as here, only captures low details about the object surfaces. Similarly, Figure 7 shows the sampling on concentric spheres for the same object, we only show the cases of clean and outliers corrupted object in the first and the second row, respectively. Figure 7-b shows the function $f(\varphi, \theta)$ sampled from sphere number 3. As can be seen, outliers appear as small noise, which is also canceled out, as can be seen in c due to the use of low frequencies. Thus, based on those results, using concentric spheres should provide better robustness to outliers.

C. CLASSIFICATION IN FREQUENCY DOMAIN

Initial spherical harmonics [13], [14] or Fourier [57], [58] based classifiers used the magnitudes of the coefficients to identify similarities between images. Those magnitudes were used because they are rotation invariant [13], have low dimensions and importantly they are useful for building robust classification techniques [57], [58]. The noise mitigation property was achieved by discarding the coefficients that are greatly affected by noise. For instance, in [58], only frequencies with high magnitudes were used, while in [57], only low-frequency components were used.

Several papers investigated the effect of noise on the Fourier coefficients [58]–[60]. In [58], the authors showed that for a given image that is perturbed with zero-mean normally distributed noise, its corresponding Fourier coefficients have the form:

$$E[|\hat{g}(n)|^2] = |\hat{f}(n)|^2 + |M|\sigma^2 \tag{8}$$

where $\hat{f}(n)$ is the n-th Fourier coefficient of the original image, $E[x]$ is the expected value of x, $\hat{g}(n)$ is the n-th Fourier coefficient for the noisy image, $|M|$ is the total number of pixels in the image, and σ is the standard deviation of the additive noise component. According to Equation 8, a particular coefficient is a useful feature for a classifier only if $|\hat{f}(n)|^2$ is much greater than $|M|\sigma^2$, or if the difference between $|\hat{g}(n)|^2$ and $|\hat{f}(n)|^2$ is very small. Although Equation 8 is derived for Fourier coefficients, its application for the spherical harmonics coefficients is straightforward as the spherical harmonics are an extension to the Fourier transform. As such, we would expect $\hat{f}(l, m)$ to be useful if $|\hat{f}(l, m)|^2 \gg |M|\sigma^2$.

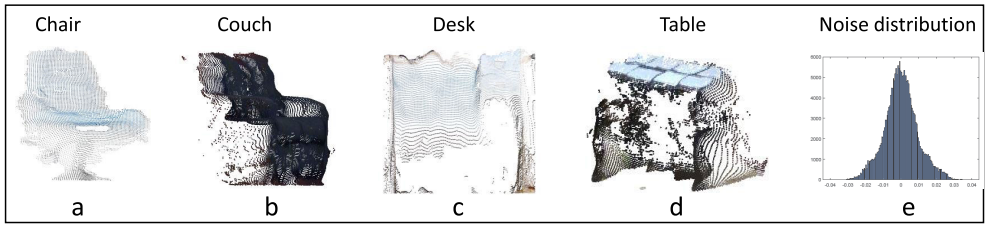

FIGURE 4. a–d: Samples of the recorded objects using Intel RealSense 3D scanner. e: Noise distributing of the captured signal.

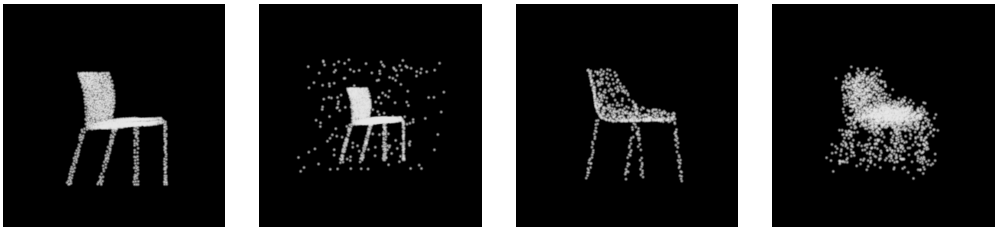

FIGURE 5. (a) The point cloud of a chair taken from the ModelNet40 dataset, (b) the same chair is corrupted with scattered outliers, (c) the same chair is corrupted with random point dropout, and (d) the same chair is perturbed with Gaussian noise.

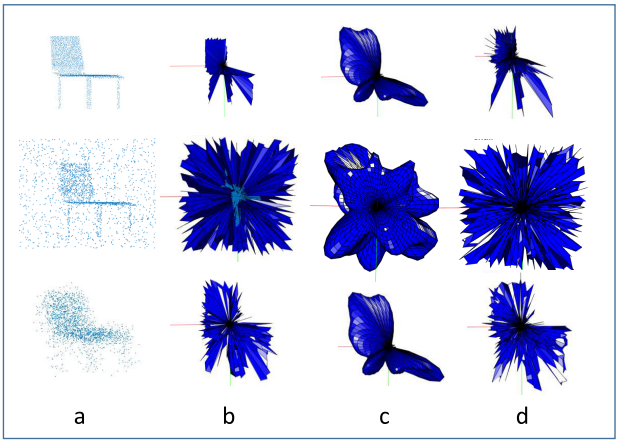

FIGURE 6. Sampling of an object (chair) using image casting, where the original object is shown in a, with its corresponding spherical function $f(\varphi, \theta)$ shown in b. The generated shape after applying Fourier transform (Equation 3) followed by inverse transform (Equation 2) is shown in (c), and the reconstruction error between b and c is shown in (d). The second row shows the same object corrupted with outliers, and the third row shows the same object perturbed with noise. All figures are in Cartesian coordinates.

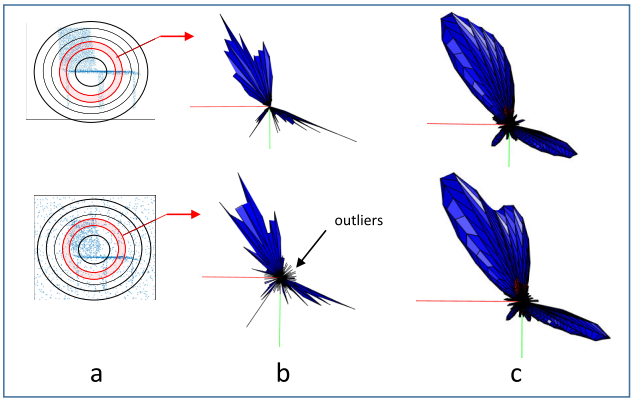

FIGURE 7. a: Sampling of an object (chair) using concentric spheres. b: The third concentric sphere corresponding spherical function $f(\varphi, \theta)$. c: The generated shape after applying Fourier transform (Equation 3) followed by inverse transform (Equation 2). The second row shows the same object corrupted with outliers. All figures are in Cartesian coordinates.

Based on [57], [58] results, we also limit the order of the spherical harmonics to suppress the effect of noise. Our experiments show that for a typical object (e.g., the chair in Figure 8) that is perturbed with Gaussian noise, the difference between the Fourier coefficients of the clean and noisy image increases by increasing the order of the spherical harmonic. As such, limiting spherical harmonics order helps reducing the effect of noise. In our classifier, only coefficients up to order 9 were used as further reducing the order will reduce the classification accuracy, as shown in our ablation study (see section IV-J). This behaviour is also seen for outliers as shown in Figure 8. The figure shows that as the degree level goes higher, outliers start to manifest themselves in the reconstructed shapes. Since we limit the use of coefficients to the orders below 10 ($l = 9$), outliers are hardly visible in Figure 8-c.

D. IMPLEMENTATION OF SPHERICAL CONVOLUTION

We propose to apply the spherical convolution on 3D models that are decomposed into concentric spheres. The use of concentric spheres generates a uniform spherical voxel grid that enables the spherical convolution neural network to learn features over the unit ball (as opposed to only learning over the unit sphere). We use separate convolution operations at each concentric sphere to allow our network to learn features relevant to that sphere. To achieve the spherical convolution, we use Equation (6) in which the learned kernel is a zonal ($m = 0$) filter h with dimension of $l \times c$, where l is the

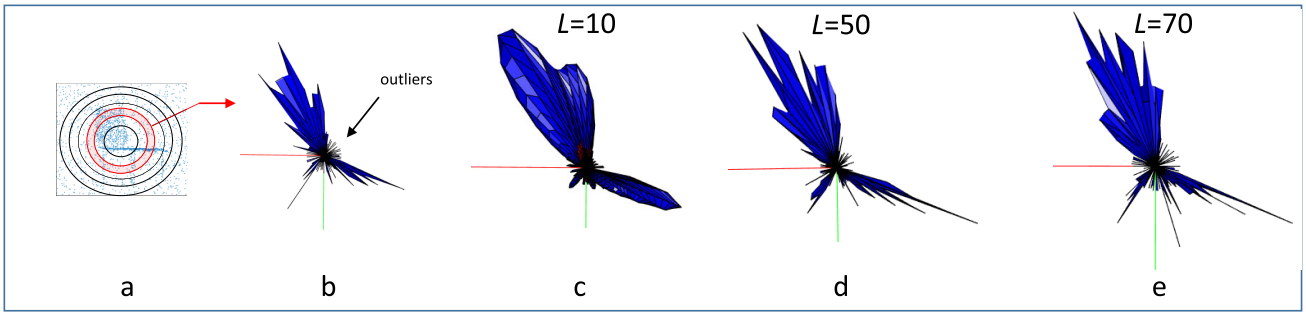

FIGURE 8. a: Sampling of an object (chair) using concentric spheres. b: The third concentric sphere corresponding spherical function $f(\varphi, \theta)$. c: The generated shape after applying inverse transform for up to degree 10. d: The generated shape after applying inverse transform for up to degree 50. and e: The generated shape after applying inverse transform for up to degree 70.

frequency degree, and c is the number of concentric spheres. Similar to [10], we parameterize the kernel filters in the spectral domain. No inverse Fourier transform is applied after the spherical convolution. Therefore, our convolution operation is entirely in the spectral domain, which reduces the convolution computation time.

Our results show that applying inverse Fourier transform (IFT) diminishes the robustness to outliers and noise as the overall accuracy reduces by more than 20 percent. Applying IFT takes us back to the input domain where it's difficult to distinguish similarity between two signals compared to the Fourier domain. This can be related to Equation (7), where for each θ and φ, the output signal is calculated by summing the entire coefficients. Thus, if the coefficients are already altered by outliers, the output signal error will be magnified/accumulated due to this summation. A detailed discussion on this topic is provided in Appendix 1. Another reason could be due to the reconstruction error shown in Figure 6-c where IFT contribute to its increase.

Our experiments show that applying the convolution operation works well with perturbed data and the network has been able to learn better features and be more discerning in terms of object classification compared to the experiments shown in Table 7. This is demonstrated by applying t-sne [61] to clean and perturbed data, and the results are provided in Appendix 1. As the application of convolutions on 3D voxel grids has shown to be robust to the influence of outliers [48], we would expect our method to exhibit a high degree of robustness to outliers as well.

Compared to previous approaches, unlike other networks such as PointNet [2] where their max-pooling chooses outliers as max, our proposed method does not use pooling or grid altering operations. The used convolution operation can be described as follows: Let $x \in X$ be our input spherical coefficient at a given degree l_i $(\hat{f}_{ij}, j \rightarrow 0 : m, m < l_i)$, the spherical convolution operation in Equation (6) is simply $f(x) = k \times (x \times \hat{h}_i)$, where $\hat{h}_i$ is the kernel value at that degree l_i and k is a constant calculated from the square root term of the same equation (Equation (6)) followed by the non-linearity operation. This mathematical operation does not alter the input signal and only assists with extracting better features in both clean and perturbed data, as seen from the t-sne results (provided in Appendix 1).

Compared to 3D CNNs such as the octnet [43], spherical CNNs are rotation equivariant, which could help in increasing our performance. In addition, the use of spherical convolution has shown to have less trainable parameters, where one layer is enough to achieve good performance [36].

E. CLASSIFICATION LAYER

The returned feature map by the convolution operations represents the feature vector defined in Equation (4). The map is then fed to fully connected and classification layers. The feature vector in Equation (5) could be used as well; however, it is less robust to data inaccuracies, as will be shown in the ablation section. Although the feature vector defined in Equation (4) is not rotation invariant, given that we are training with rotations, we would expect our network to learn rotations.

IV. EXPERIMENTS

We compare our framework with state-of-the-art published spherical convolution architectures, point cloud classification methods, and robust methods. We considered outliers, noise, and missing points as our types of data inaccuracies in this paper since the corruption of point clouds with such inaccuracies is common.

A. DATASETS

To test the robustness of our approach and other methods, we use the benchmarks ModelNet40 [30], [42], ScanObjectNN [55], and shapenet [62] datasets. We also build a small dataset that contains 56 scenes of some ScanObjectNN objects captured by Inter RealSense scanner. In addition, we used MNIST in the supplementary materials. We generate three instances from each of the test sets of ModelNet40, ScanObjectNN, and shapenet. Each of these instances is either corrupted with outliers, corrupted with missing points, or perturbed with noise. We report the classification accuracy on each copy individually along with the classification accuracy on the original test set. i.e., for the ModelNet40 dataset, which has a test set of 2468 objects,

we report the classification accuracy on the original test set, the original test that was perturbed with noise, the original test that was corrupted with outliers, and the original test that was corrupted with missing points (each test set has the 2,468 objects). Figure 5 shows a sample of those perturbations and corruptions. For MNIST dataset, we perturb the test set with random noise/outliers at different ratios. The details of these perturbations and corruptions are explained in the following sections.

B. ARCHITECTURE

The proposed architecture, shown in Figure 3, works with voxel-based objects. As such, a given 3D shape needs to be converted to a 3D voxel by dividing the space into a 64*64*7 grid as shown in Figure 3-A. The number of concentric spheres is chosen to be c=7 (see Figure 13) with a grid resolution of 64 by 64 for each sphere. The Fourier transform (Equation 3) is applied on each of the seven concentric spheres to get the spherical harmonics coefficients $\hat{f}_l^m$ with $l = 9$ as the degree of the spherical harmonics. Next, the spherical convolution (Equation 6) is applied to the spherical harmonics coefficients. We used one convolution layer (with a size $l = 9$) having 16 output channels, and used relu after the convolution operation as our non-linearity. The output of the convolution layer is then fed into a fully connected layer with a size of 1024, followed by a classification layer. The spherical convolution kernel $\hat{h}_l^0$ applied on each sphere is a zonal filter with a size of 1 by 9. The spherical convolution operation is equivalent to the inner product between a matrix with a size of 9 by 9 that contains the spherical harmonics coefficients and a vector of length 9 that represents the convolution kernel. We compared different architectures in the ablation study.

C. TRAINING

We perform data augmentation for training by including random rotations around the vertical axis (between $0 - 2\pi$) and small jittering (0.01 Gaussian noise). We take into account points normal's in some scenarios (we mention those scenarios when we report the classification accuracy). The patch size is set to 16, the learning rate varies from 0.001 to 0.00004, and the number of epochs is set to 48. We used a TITAN Xp GPU, where only 450Mb of memory was used during training.

D. CLASSIFICATION PERFORMANCE ON OUR CAPTURED SCENES

The classification accuracy of the proposed method and state-of-art methods on our datasets are shown in Table 1. The dataset contains 56 objects that belong to five categories from the ScanObjectNN dataset (chairs, tables, cabinets, sofas, and desks). We train all methods on the ScanObjectNN training data while we test them on our recorded objects. Our proposed model scores 75% classification accuracy on the captured scenes, while PointNet [2] and pointCNN [63] score 66% classification accuracy. DGCNN [6] and KPConv [64] score around 53%, while PL-Net3D [7] scores 57%. Each category in our dataset contains scenes with different noise levels. As such, methods can identify objects up to a certain noise level. Comparing the results presented in Table 1 show that our method can identify objects at higher noise levels compared to other methods.

TABLE 1. The classification accuracy of some state-of-art methods on our captured scenes.

Method	Accuracy
PL-Net3D [7]	57
PointNet [2]	66
DGCNN [6]	51.7
PointCNN [63]	66
KPConv [64]	53.6
RSCNN (**ours**)	**75**

E. ROBUSTNESS TO OUTLIERS

In this section, we take the ModelNet40 test dataset and corrupt it with outliers. Similar to [7], we present two outlier scenarios generated with different mechanisms. In the first scenario, we test our model in the presence of scattered outliers: points uniformly distributed in the unit cube. A sample case is shown in Figure 5. In the second scenario, added outliers are grouped into clusters of ten or twenty points, which are uniformly distributed in the unit cube (similar to [65]). The overall number of scattered points for this scenario are fixed to ten or twenty percent as shown in Table 2. Points in each cluster are normally distributed with zero mean and standard deviations of 4% and 6%.

In Figure 9, we show the inference time and GPU memory usage for SPHnet, octnet, PL-Net3D and our method. The shown inference time include the preprocessing time required for SPHnet, octnet, and our method to convert the point cloud to voxels, and the preprocessing time for PL-Net3D to detect all the planes in the point cloud using RANSAC. As can be seen from the figure, our method is faster than any other compared method, while the iterative RANSAC in PL-Net3D takes 100 times longer to detect all the planes in the point cloud. The figure also show that our method uses less GPU memory than other compared methods.

Figure 10 and Table 2 show that our model is highly robust to the influences of outlier in both scenarios. The classification accuracy only drops by 8% percent when half the data are outliers. We get similar robustness to PL-net3D with the benefit of being much faster (100 times faster), while other models robustness drop by significantly higher margins. For Spherical-cnn [10], even when we used the median aggregation for generating the unit sphere grid (instead of max aggregation), the network remains sensitive to the influences of the outliers. Similarly, SpH-net [12] performs poorly when there were outliers as these outliers distort the distance graph.

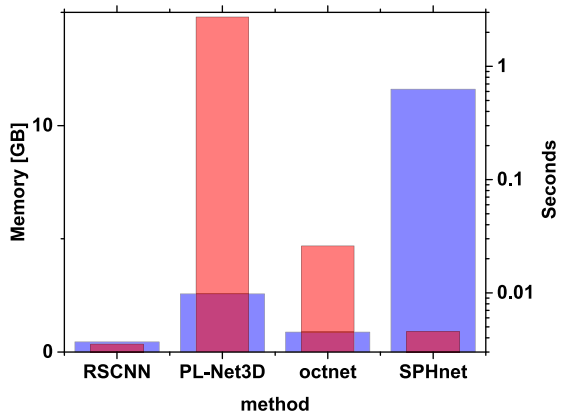

FIGURE 9. GPU memory usage (blue/wide columns) along with the inference time in seconds (red/thin columns, in log scale) for SPHnet, octnet, PL-Net3D and our method (RSCNN). The shown times include the preprocessing time required for SPHnet, octnet, and our method to convert the point cloud to voxels, and the preprocessing time for PL-Net3D to detect all the planes in the point cloud.

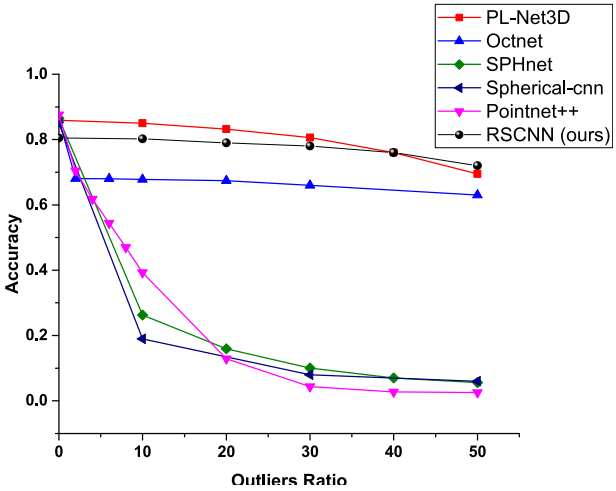

FIGURE 10. Classification accuracy versus outliers.

F. ROBUSTNESS TO NOISE

In this section, we use the ModelNet40 dataset and add noise to object points. We simulated the effect of noise in point cloud data by perturbing points with zero mean, normally distributed values with standard deviations ranging from 0.02 to 0.10. A sample case is shown in Figure 5. We used Gaussian noise as we found out that noise in real-scene objects is relatively Gaussian as shown in Figure 4. Figure 11 show that our proposed model performance deteriorated the least compared to other models (by around 18%) for relatively large amount of noise (at 0.10 noise level). This can be related to the use of low frequencies, as we mentioned earlier in Figure 6, which cancels the effect of noise. SPHnet was significantly affected by noise, while spherical-CNN performance was relatively much better than SPHnet.

G. ROBUSTNESS TO MISSING POINTS

In this section, we use the ModelNet40 dataset and randomly remove points from each object. Figure 12 shows that our

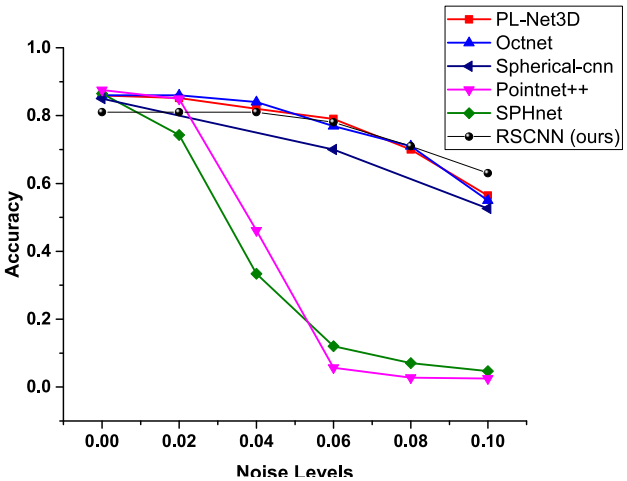

FIGURE 11. Classification accuracy versus noise.

TABLE 2. Object classification results on clustered outliers.

Method	10% 10p $\mathcal{N}(0.04)$	10% 10p $\mathcal{N}(0.06)$	20% 20p $\mathcal{N}(0.04)$
PointNet [2]	7	6	7
SPHnet [12]	22	24	8
KPConv [64]	13	11	7
Octnet [43]	47	48	37
PL-Net3D [7]	**79**	**80**	67
RSCNN (ours)	**79**	79	**73**

10%: outliers percentage, 10p: 10 Points in each cluster.

model classification accuracy drops by only 2% when half the points are eliminated, and by 22% when 90% of points are removed. spherical-cnn classification accuracy drops by 10% when half the points are eliminated and it degrades after that. SPHnet classification accuracy drops by 8% when 60% of points are eliminated and it degrades after that. Ocntnet classification accuracy degrades after 50%.

The above results are summarized in Table 3 below. Our proposed model scores 82.2% classification accuracy on ModelNet40 (MN40) dataset when using points normals, while we achieve 80.5% with points only. Point-based methods such as DGCNN [6] and KPConv [64] score around 92% classification accuracy. However, when testing those methods on MN40 corrupted with outliers, noise, and missing points, our proposed model scores the highest classification performance.

H. CLASSIFICATION ON SHAPENET DATASET

Shapenet dataset contains 51,127 pre-aligned shapes from 55 categories, which are split into 35,708 for training, 5,158 shapes for validation and 10,261 shapes for testing. Each object contains 2048 points normalized in the unit cube.[1] We tested several methods on this dataset and the results are shown in Table 4. PointCNN achieves the highest

[1] https://github.com/AnTao97/PointCloudDatasets

TABLE 3. The classification accuracy on ModelNet40 (MN40) dataset. Noise, Dropout, and OUT are the same dataset when its corrupted with 0.1 Gaussian noise, 90% missing points, and 50% outliers respectively.

Method	Input	MN40	OUT	Noise	Dropout
Spherical-cnn [10]	image	87	5	53	27
PL-Net3D [7]	Points	86.6	70	60	50
PointNet [2]		89	4	27	57
DGCNN [6]		**92.2**	5	5	18
PointNet++ [30]		91.8	2	2	30
PointCNN [63]		91	20	4	7
KPConv [64]		90	4	4	12
SPHnet [12]		91	5	5	5
OctNet [43]	Voxels	86.5	63	55	25
VoxNet [41]		86	7	17	28
RSCNN (**ours**)		82.2 (80.5)	**72**	**63**	**58**

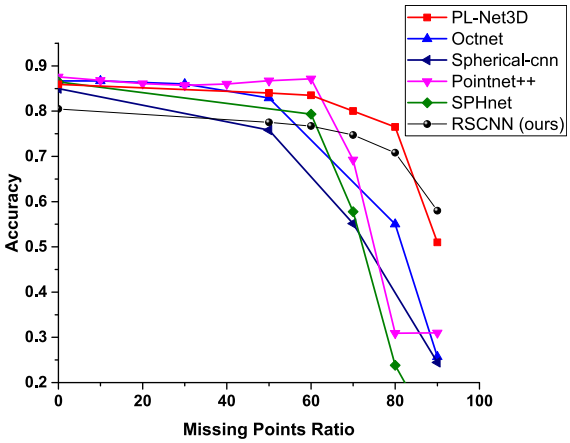

FIGURE 12. Classification accuracy versus missing points.

classification accuracy with a score of 83%, whereas PointNet and DGCNN score around 82%, while KPConv and VoxNet score around 81%. Our proposed model scores 77.4% using points and their normals (75.6% with points only), which is around 3% less than VoxNet and 5% less than the best model.

While the performance of the proposed model on clean data is slightly lower, it shows significant robustness on the corrupted datasets as seen in the table. Pl-Net3D outperforms our method by 7% on objects corrupted with 50% outliers, however, our model outperforms Pl-Net3D on data corrupted with noise and missing points by 12% and 1% respectively.

I. CLASSIFICATION PERFORMANCE ON ScanObjectNN DATASET

We corrupt the ScanObjectNN (SC) dataset with 50% outlier, 0.1 Gaussian noise, and 80% missing points. We then report the classification accuracy of our proposed model along with some state-of-art models on the corrupted ScanObjectNN (SC) in Table 5. Our proposed model scores 76% and 76% classification accuracies on the original ScanObjectNN (SC) datasets with and without points normal's respectively, while we score the highest classification performance when data are corrupted with noise or outliers. KPConv scores the best classification accuracy with a value of 89%, followed by PointCNN with a value of 87%.

Comparing the performance of all methods on ScanObjectNN and ModelNet40 datasets shows that all methods classification accuracies (including ours) drop by 4-6%. This could be due to the lower number of training data of ScanObjectNN compared to ModelNet40.

PL-Net3D scores 70% classification accuracy with lower robustness to noise, outliers, and missing points. Although KPConv [64] scores the highest classification accuracy on the ScanObjectNN dataset. However it shows low robustness to outliers, noise, and random point dropout along with all other compared methods, except that PointNet shows better robustness to missing points.

J. ABLATION STUDY

We conducted several experiments on the ModelNet40 dataset to explore all possible solutions of our method and their performance under different data inaccuracies; the results are shown in Table 6. The first row shows the results of our proposed model (RSCNN) with 3D Models having 2000 points and their normals. The second row shows the results of our proposed model with points only and normalizing inputs to get a density grid (same results shown in the previous section). The third row shows the results when training without normalizing inputs. As can be seen from those results, using density grid provides the best performance. The forth row shows the result of our proposed model trained without points jittering (using points only with normalizing inputs).

We implemented the inverse transform operation after applying the convolution in our model. As a result, our model performed worse and became less robust to data inaccuracies. The outputs are presented in the fifth row of Table 6. The effect of performing Inverse Fourier Transform on the classification accuracy is discussed in Supplement 1. In the next

TABLE 4. The classification accuracy on Shapenet dataset.

Method	Input	clean	OUT	Noise	Dropout
PointNet	Points	82.2	4	46	52
PointCNN		**83**	2	2	19
DGCNN		82.3	4	13	17
KPConv		81.2	2	2	8
PL-Net3D		78	**66**	47	58
VoxNet	Voxels	80.9	19	31	14
RSCNN (**ours**)		77.4 (75.6)	59	**58**	**59**

TABLE 5. The classification accuracy on ScanObjectNN dataset.

Method	Input	SC	OUT	Noise	Dropout
PL-Net3D [7]	Points	70	20	40	55
PointNet [2]		82	8	26	**80**
DGCNN [6]		85	18	20	17
PointCNN [63]		87	44	22	25
KPConv [64]		**89**	8	14	68
VoxNet [41]	Voxels	80	17	20	20
RSCNN (**ours**)		76 (74)	**58**	**55**	70

TABLE 6. Classification accuracy versus different network architectures and different data inaccuracies.

method	sampled points	classification accuracy			
		clean	90% dropout	0.1 noise	50% outliers
RSCNN	2k+N	**82.2**	39	48.2	56
RSCNN	2k	80.5	58	**63**	**72**
RSCNN no NL	2k	80.7	30	55	70
RSCNN*	2k	80.8	**61**	58	70
RSCNN + IFT	2k	65	58	29	21
RSCNN no FC	2k	66	43	51	44

TABLE 7. Classification accuracy results for objects perturbed with noise, missing points, and outliers.

occupancy grid	clean	random dropout	noise	outliers
binary+ D1	**0.79**	0.34	0.24	0.14
density+ D1	0.78	**0.75**	**0.37**	**0.50**
density+ D2	0.68	0.68	0.3	0.24

TABLE 8. Effect of order level on the relative difference $(|\hat{g}(l,m)|^2 - |\hat{f}(l,m)|^2)/|\hat{f}(l,m)|^2$ between coefficients of noisy and clean image.

Order	Mean	Median	Std
9	9	0.04	60
15	260	0.065	5859
20	2202	0.07	59160
60	71660	0.05	6663551

step, we evaluated our model with no fully connected layer to reduce the number of trainable parameters. However, the results, shown in the sixth row, suggest that such an action is detrimental for the overall performance.

We evaluated the robustness of the two descriptors represented by Equation (4) and Equation (5) for object classification by feeding each of them to a fully connected neural network. The results are shown in Table 7. We tested their robustness against: Gaussian noise with 0.10 standard deviations, uniformly scattered outliers with 50% percentage, and 80% Random point dropout. The results show that the used sampling and the density occupancy grid provide a high degree of robustness to outliers. In addition, these results also show that the descriptor in Equation (4) D1 provides higher classification accuracy than using the descriptor in Equation (5) D2 as the first one carries more shape information.

We tested our method with 5, 7, and 10 concentric spheres. Each sphere had a 64 by 64 grid. We also tested our method

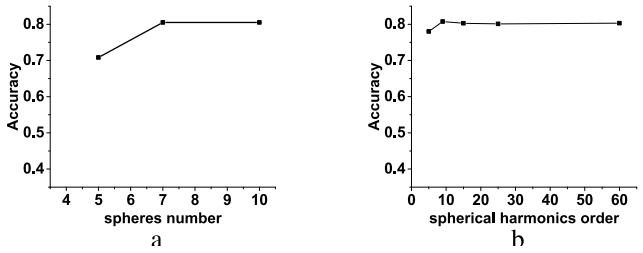

FIGURE 13. (a) The Classification accuracy versus number of concentric spheres. (b) The Classification accuracy versus the spherical harmonics order *l*.

with spherical harmonics orders ranging from 9 to 60. The results are shown in Figure 13. Our network performance gradually increases up to using 7 concentric spheres and plateaus afterwards. Moreover, increasing the spherical harmonics order did not improve the accuracy.

Table 8 shows that for a typical object (e.g., the chair in figure 8) that is perturbed with 2% Gaussian noise, the difference between the Fourier coefficients of the clean and noisy image increases by increasing the order of the spherical harmonic. As such, limiting spherical harmonics order helps reducing the effect of noise. In our classifier, only coefficients up to order 9 were used as further reducing the order will reduce the classification accuracy, as shown in Figure 13-b.

V. CONCLUSION

Classifying 3D objects is an important task in several robotic applications. In this paper, we present a robust spherical harmonics model for single object classification. Our model uses the voxel grid of concentric spheres to learn features over the unit ball. In addition, we keep the convolution operations in the Fourier domain without applying the inverse transform used in previous approaches. As a result, our model is able to learn features that are less sensitive to data inaccuracies. We tested our proposed model against several types of data inaccuracies, such as noise and outliers. Our results show that the proposed model competes with the state-of-art networks in terms of robustness to effects of data inaccuracies with lower computational requirements.

REFERENCES

[1] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3D LiDAR sensor," *Robot. Auton. Syst.*, vol. 59, no. 5, pp. 265–273, 2011.
[2] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
[3] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
[4] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
[5] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.
[6] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[7] A. Mukhaimar, R. Tennakoon, C. Y. Lai, R. Hoseinnezhad, and A. Bab-Hadiashar, "PL-Net3D: Robust 3D object class recognition using geometric models," *IEEE Access*, vol. 7, pp. 163757–163766, 2019.
[8] D. Bulatov, D. Stütz, J. Hacker, and M. Weinmann, "Classification of airborne 3D point clouds regarding separation of vegetation in complex environments," *Appl. Opt.*, vol. 60, no. 22, pp. F6–F20, 2021.
[9] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling, "Spherical CNNs," 2018, *arXiv:1801.10130*.
[10] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) equivariant representations with spherical CNNs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 52–68.
[11] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications," *Astron. Comput.*, vol. 27, pp. 130–146, Apr. 2019.
[12] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, and M. Ovsjanikov, "Effective rotation-invariant point CNN with spherical harmonics kernels," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 47–56.
[13] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Process.*, vol. 6, 2003, pp. 156–164.
[14] D. Wang, S. Sun, X. Chen, and Z. Yu, "A 3D shape descriptor based on spherical harmonics through evolutionary optimization," *Neurocomputing*, vol. 194, pp. 183–191, Jun. 2016.
[15] R. Green, "Spherical harmonic lighting: The gritty details," in *Proc. Arch. Game Developers Conf.*, vol. 56, 2003, p. 4.
[16] C. R. Nortje, W. O. C. Ward, B. P. Neuman, and L. Bai, "Spherical harmonics for surface parametrisation and remeshing," *Math. Problems Eng.*, vol. 2015, pp. 1–11, Jan. 2015.
[17] T. Bülow and K. Daniilidis, "Surface representations using spherical harmonics and Gabor wavelets on the sphere," CIS, Minsk, Belarus, Tech. Rep. MS-CIS-01-37, 2001, p. 92.
[18] K. Yücer, A. Sorkine-Hornung, O. Wang, and O. Sorkine-Hornung, "Efficient 3D object segmentation from densely sampled light fields with applications to 3D reconstruction," *ACM Trans. Graph.*, vol. 35, no. 3, pp. 22:1–22:15, Jun. 2016.
[19] M. Bijelic, T. Gruber, and W. Ritter, "A benchmark for LiDAR sensors in fog: Is detection breaking down?" in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 760–767.
[20] Y. Li, P. Duthon, M. Colomb, and J. Ibanez-Guzman, "What happens for a ToF LiDAR in fog?" *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6670–6681, Nov. 2021.
[21] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary LiDAR and radar signals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 444–453.
[22] A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias, "Quantifying the influence of rain in LiDAR performance," *Measurement*, vol. 95, pp. 143–148, Jan. 2017.
[23] C. Goodin, D. Carruth, M. Doude, and C. Hudson, "Predicting the influence of rain on LiDAR in ADAS," *Electronics*, vol. 8, no. 1, p. 89, Jan. 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/1/89
[24] X. Cheng, Y. Zhong, Y. Dai, P. Ji, and H. Li, "Noise-aware unsupervised deep LiDAR-stereo fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6339–6348.
[25] X. Wang, Z. Pan, and C. Glennie, "A novel noise filtering model for photon-counting laser altimeter data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 947–951, Jul. 2016.
[26] A. H. Incekara, D. Z. Seker, and B. Bayram, "Qualifying the LiDAR-derived intensity image as an infrared band in NDWI-based shoreline extraction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5053–5062, Dec. 2018.
[27] Z. Song, H. Zhu, Q. Wu, X. Wang, H. Li, and Q. Wang, "Accurate 3D reconstruction from circular light field using CNN-LSTM," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.
[28] D. Dimou and K. Moustakas, *Fast 3D Scene Segmentation and Partial Object Retrieval Using Local Geometric Surface Features*. Berlin, Germany: Springer, 2020, pp. 79–98, doi: 10.1007/978-3-662-61364-1_5.
[29] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–73, Jul. 2013.
[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[31] J. R. Driscoll and D. M. J. Healy, "Computing Fourier transforms and convolutions on the 2-sphere," *Adv. Appl. Math.*, vol. 15, no. 2, pp. 202–250, 1994.

[32] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5028–5037.

[33] M. Weiler, F. A. Hamprecht, and M. Storath, "Learning steerable filters for rotation equivariant CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 849–858.

[34] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, "3D steerable CNNs: Learning rotationally equivariant features in volumetric data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10381–10392.

[35] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," 2018, *arXiv:1803.10091*.

[36] S. Ramasinghe, S. Khan, N. Barnes, and S. Gould, "Representation learning on unit ball with 3D roto-translational equivariance," *Int. J. Comput. Vis.*, vol. 128, no. 6, pp. 1–23, 2019.

[37] W. Boomsma and J. Frellsen, "Spherical convolutions and their application in molecular modelling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3433–3443.

[38] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 529–539.

[39] B. Coors, A. Paul Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2018, pp. 518–533.

[40] Y. You, Y. Lou, Q. Liu, Y.-W. Tai, W. Wang, L. Ma, and C. Lu, "PRIN: Pointwise rotation-invariant network," 2018, *arXiv:1811.09361*.

[41] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[42] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[43] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.

[44] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," 2016, *arXiv:1608.04236*.

[45] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3813–3822.

[46] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," 2019, *arXiv:1906.01592*.

[47] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3145–3152, Oct. 2018.

[48] A. Mukhaimar, R. Tennakoon, C. Y. Lai, R. Hoseinnezhad, and A. Bab-Hadiashar, "Comparative analysis of 3D shape recognition in the presence of data inaccuracies," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2471–2475.

[49] S. Gould, R. Hartley, and D. Campbell, "Deep declarative networks: A new hope," 2019, *arXiv:1909.04866*.

[50] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578.

[51] Y. Xu, S. Arai, F. Tokuda, and K. Kosuge, "A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color," *IEEE Access*, vol. 8, pp. 70262–70269, 2020.

[52] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.

[53] B. Douillard, J. Underwood, V. Vlaskine, A. Quadros, and S. Singh, "A pipeline for the segmentation and classification of 3D point clouds," in *Experimental Robotics*. Berlin, Germany: Springer, 2014, pp. 585–600.

[54] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 4034–4041.

[55] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[56] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[57] R. Maani, S. Kalra, and Y.-H. Yang, "Noise robust rotation invariant features for texture classification," *Pattern Recognit.*, vol. 46, no. 8, pp. 2103–2116, Aug. 2013.

[58] S. Hui and S. H. Żak, "Discrete Fourier transform based pattern classifiers," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 62, no. 1, pp. 15–22, Mar. 2014.

[59] R. Bardenet and A. Hardy, "Time-frequency transforms of white noises and Gaussian analytic functions," *Appl. Comput. Harmon. Anal.*, vol. 50, pp. 73–104, Jan. 2021.

[60] J. Huillery, F. Millioz, and N. Martin, "Gaussian noise time-varying power spectrum estimation with minimal statistics," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5892–5906, Nov. 2014.

[61] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[62] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[63] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 820–830.

[64] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[65] X. Ning, F. Li, G. Tian, and Y. Wang, "An efficient outlier removal method for scattered point cloud data," *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0201280.