Supplementary Material for RISurConv: Rotation Invariant Surface Attention-Augmented Convolutions for 3D Point Cloud Classification and Segmentation

Zhiyuan Zhang[1], Licheng Yang[2], and Zhiyu Xiang[2]

[1] School of Computing and Information Systems, Singapore Management University
https://github.com/cszyzhang/RISurConv
[2] College of Information Science and Electronic Engineering, Zhejiang University

Abstract. In this document, we provide the mathematical proof for the completeness of our Rotation Invariant Surface Properties (RISP) defined in the main paper. Additionally, We show the output dimensions of each layer of the proposed neural network and explain the deconvolution layer in more details. The performance of the proposed method on other baselines is also investigated. We also evaluate our method on another large-scale semantic segmentation S3DIS dataset. Finally, we look into the classification performance by comparing the per-class accuracies with the state-of-the-art methods.

1 Completeness Proof of RISP

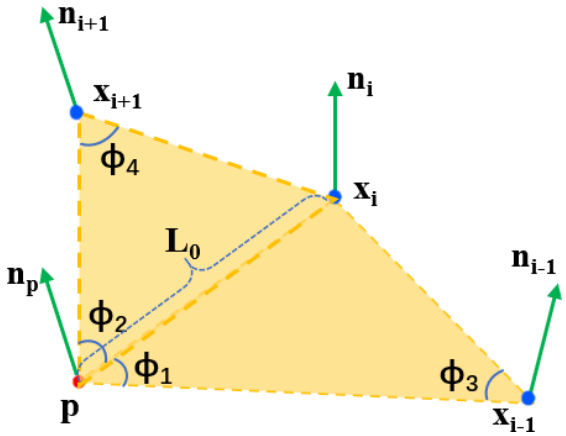

Fig. 1: RISP for two triangles. p is the reference point, while x_i, x_{i-1} and x_{i+1} are the two neighbors. n_p, n_i, n_{i-1} and n_{i+1} are the corresponding normal vectors located at p, x_i, x_{i-1}, and x_{i+1} respectively.

In this section, we provide the completeness proof of RISP comprising 14 properties defined in the main paper. To facilitate illustration, we write down the RISP Equation here again:

$$\begin{aligned} \mathrm{RISP}(x_i) = [&L_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \\ &\alpha_1, \alpha_2, \beta_1, \beta_2, \theta_1, \theta_2, \gamma_1, \gamma_2] \end{aligned} \tag{1}$$

where L_0 is the distance from reference p to neighbor x_i, while ϕ_1 to ϕ_5 measure the two triangles as well as the relationship between the two triangle surfaces with regard to the edge $\overrightarrow{px_i}$ in the Euclidean space:

$$\begin{aligned} \phi_1 &= \angle\left(\overrightarrow{x_{i-1}p}, \overrightarrow{x_ip}\right), \\ \phi_2 &= \angle\left(\overrightarrow{x_{i+1}p}, \overrightarrow{x_ip}\right), \\ \phi_3 &= \angle\left(\overrightarrow{x_{i-1}x_i}, \overrightarrow{x_{i-1}p}\right), \\ \phi_4 &= \angle\left(\overrightarrow{x_{i+1}p}, \overrightarrow{x_{i+1}x_i}\right), \\ \phi_5 &= \angle\left(\overrightarrow{x_{i+1}p} \times \overrightarrow{x_ip}, \overrightarrow{x_{i-1}p} \times \overrightarrow{x_ip}\right). \end{aligned} \tag{2}$$

while other properties describe the two surfaces in the tangent space, e.g. normal vectors can define the directions in which the surface is bending away from the tangent space:

$$\begin{aligned} &\alpha_1 = \angle\left(\overrightarrow{n_p}, \overrightarrow{x_ip}\right), \alpha_2 = \angle\left(\overrightarrow{n_p}, \overrightarrow{x_{i-1}p}\right), \\ &\beta_1 = \angle\left(\overrightarrow{n_i}, \overrightarrow{x_ip}\right), \beta_2 = \angle\left(\overrightarrow{n_i}, \overrightarrow{x_{i-1}x_i}\right), \\ &\theta_1 = \angle\left(\overrightarrow{n_{i-1}}, \overrightarrow{x_{i-1}p}\right), \theta_2 = \angle\left(\overrightarrow{n_{i-1}}, \overrightarrow{x_{i-1}x_i}\right), \\ &\gamma_1 = \angle\left(\overrightarrow{n_{i+1}}, \overrightarrow{x_{i+1}x_i}\right), \gamma_2 = \angle\left(\overrightarrow{n_{i+1}}, \overrightarrow{x_{i+1}p}\right). \end{aligned} \tag{3}$$

To prove the completeness of these 14 properties, we start from single triangle analysis, and then extend to two triangles.

1.1 Completeness of RISP for One Triangle

Take Fig. 2 for instance, L_0 of Eq. (1) with ϕ_1 and ϕ_3 of Eq. (2) can uniquely solve a triangle. We prove this statement as follows.

Given the fact that the sum of the angles in a triangle is always 180 degrees, the third angle can always be determined.

Based on the Law of Sines formula, we can find the lengths of the other two sides of the triangle. The Law of Sines states that in any triangle ABC with sides a, b, and c and opposite angles A, B, and C, the following relationship holds:

$$\frac{a}{sin(A)} = \frac{b}{sin(B)} = \frac{c}{sin(C)}. \tag{4}$$

By using these formulas, we can calculate all the missing angles and side lengths of the triangle, and thus uniquely solve it. To prove the completeness of L_0, ϕ_1 and ϕ_3 for describing a triangle in Euclidean space, we need to show that any

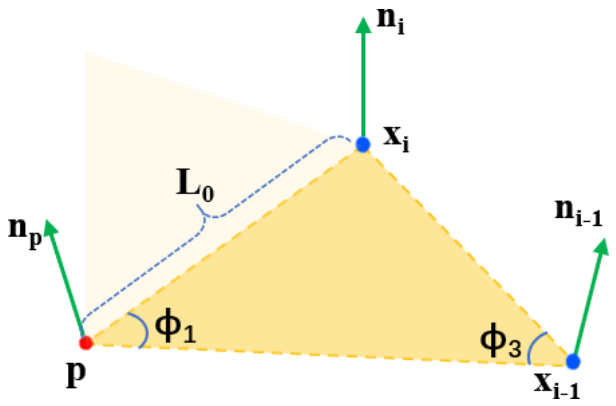

Fig. 2: RISP analysis for single triangle.

other triangle with the same angles and edge length must be congruent (i.e., they have the same shape and size).

Now, consider two triangles with the same two angles ϕ_1 and ϕ_3 and the same edge length L_0. Let b_1 and c_1 be the lengths of the other two edges of the first triangle, and let b_2 and c_2 be the lengths of the other two edges of the second triangle. We have already shown that $b_1 = b_2$ and $c_1 = c_2$, so the two triangles have the same shape.

To show the two triangles also have the same size, we can use the formula for the area of a triangle in terms of its edges:

$$A = \frac{1}{2} L_0 b \sin \phi_1 = \frac{1}{2} bc \sin \phi_3 \tag{5}$$

Since L_0, ϕ_1 and ϕ_3 are the same for both triangles, we have $A_1 = A_2$ if and only if $b_1 = b_2$ and $c_1 = c_2$. Therefore, the two triangles must be congruent. This shows that two angles and one edge length are complete for describing a triangle in Euclidean space, up to congruence.

While L_0, ϕ_1 and ϕ_3 determine the triangle in the Euclidean space uniquely, the properties in tangent space is also crucial for analysing the behaviour of the surface. The normal vector defines the direction in which the surface is bending away from the tangent space at that point. So, we use α_1, α_2, β_1, β_2, θ_1, θ_2 of Eq. (3) as the rotation invariant properties to encode the angles between the normals and the triangle edges. To prove the completeness of these 6 properties for describing the relationship between the normals and the triangle, we use θ_1 and θ_2 for illustration which encode the angles between n_{i-1} and two edges located at x_{i-1}. To prove the completeness, we need to show that any other normal vector at the same point (x_{i-1}) on the triangle can be obtained by rotating the original normal vector by some combination of the two edges of the triangle, with angles equal to θ_1 and θ_2.

Let the original normal vector be denoted by $\overrightarrow{n_{i-1}}$ for this case, and let $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ be the two edges of the triangle at the same point as $\overrightarrow{n_{i-1}}$.

Consider any other normal vector $\overrightarrow{n_{i-1}}'$ at the same point on the triangle. We can express $\overrightarrow{n_{i-1}}'$ as a linear combination of $\vec{e_1}$, $\vec{e_2}$, and $\overrightarrow{n_{i-1}}$, as follows:

$$\overrightarrow{n_{i-1}}' = a\vec{e_1} + b\vec{e_2} + c\overrightarrow{n_{i-1}} \tag{6}$$

where a, b, and c are scalar coefficients.

We can then express a, b, and c in terms of the angles between $\overrightarrow{n_{i-1}}'$ and the two edges of the triangle, θ_1' and θ_2', as follows:

$$a = \frac{\overrightarrow{n_{i-1}}' \cdot \vec{e_1}}{|\vec{e_1}||\overrightarrow{n_{i-1}}'|} = \frac{\cos\theta_1'}{\sqrt{1-\cos^2\theta_1'}}|\overrightarrow{n_{i-1}}'|$$

$$b = \frac{\overrightarrow{n_{i-1}}' \cdot \vec{e_2}}{|\vec{e_2}||\overrightarrow{n_{i-1}}'|} = \frac{\cos\theta_2'}{\sqrt{1-\cos^2\theta_2'}}|\overrightarrow{n_{i-1}}'|$$

$$c = \sqrt{1-a^2-b^2}$$

We can then express $\overrightarrow{n_{i-1}}'$ in terms of the original normal vector and the two edges of the triangle as follows:

$$\overrightarrow{n_{i-1}}' = \cos\theta_1' \frac{\overrightarrow{n_{i-1}} \times \vec{e_1}}{|\overrightarrow{n_{i-1}} \times \vec{e_1}|} + \cos\theta_2' \frac{\overrightarrow{n_{i-1}} \times \vec{e_2}}{|\overrightarrow{n_{i-1}} \times \vec{e_2}|} + \sqrt{1-\cos^2\theta_1'-\cos^2\theta_2'}\overrightarrow{n_{i-1}} \tag{7}$$

This expression shows that any other normal vector at the same point on the triangle can be obtained by rotating the original normal vector by some combination of the two edges of the triangle, with angles equal to θ_1 and θ_2. Therefore, the two angles between a normal vector at one point of the triangle and the two edges of that triangle are complete in describing the relationship between the normal vector and the triangle.

1.2 Completeness of RISP for Two Triangles

Now we prove the completeness of RISP for two triangles. As is shown in Fig. 1, by adding one more point x_{i+1}, we have two triangles sharing the edge $\overrightarrow{x_ip}$. We need more properties to encode the second triangle as well we the relationship between these two triangles.

We use L_0, ϕ_2 and ϕ_4 to uniquely determine the second triangle in the Euclidean space. We also we use ϕ_5 to encode the information about the relative orientation of the two triangles. In the tangent space, we add two properties γ_1 and γ_2 to encode the angles between n_{i+1} and the two adjacent edges.

Note that we do not explicitly encode the angle between n_i and $\overrightarrow{x_ix_{i+1}}$ and the angle between n_p and px_{i+1}, because these two angles can be computed based on the RISP properties.

The relative position of n_i to the first triangle $\triangle px_ix_{i-1}$ is already fixed by β_1 and β_2, while the relative position of the second triangle $\triangle px_ix_{i+1}$ to the first triangle $\triangle px_ix_{i-1}$ is also fixed by ϕ_5. So, the angle between n_i and x_ix_{i+1} is also fixed.

To compute the angle μ between n_i and $\overrightarrow{x_i x_{i+1}}$ for example, we can use the following law of tetrahedron to find the angle:

$$\cos\mu = \cos\phi_6 \cos\beta_1 + \sin\phi_6 \sin\beta_1 \cos\phi_5,$$

where $\phi_6 = \pi - \phi_2 - \phi_4$

Now, we have explained all the 14 properties used in Eq. (1) and proved the completeness. In the ablation study of the main paper, we also see that when the angle μ between n_i and $\overrightarrow{x_i x_{i+1}}$ is added the performance does not improve. This also verifies the completeness of RISP.

2 RISurConv Network

In Figure 3 of the main paper, we have plotted the RISurConv network architecture. To further illustrate the network details, we list the output dimensions and points for each layer of the classification and segmentation as follows in Table 1. Please note that the deconvolution process of the segmentation network follows a similar approach to RISurConv . The key distinction lies in our convolution's output, which targets a point subset with an increased number of feature channels. In contrast, deconvolution directs its output to a point set with more points compared to the input, yet with fewer feature channels. Specifically, if we consider deconvolution starting with the set of N_l points at layer l, the RISurConv operator is applied to upsample the features of these N_l points to a set of N_{l+1} points at the subsequent layer $l+1$. This deconvolution process iterates until the point cloud regains its original number of points, denoted as N.

It's important to highlight that, due to the availability of point subsets during the downsampling phase in the encoder section, there is no need to generate new points during the upsampling. Instead, we can efficiently reuse these subsets and propagate the features through interpolation.

3 Other Baselines

Our network architecture is flexible in that it can be used for different baselines by replacing the xyz input with the proposed rotation invariant properties. Use pointnet++ and DGCNN as baselines, we tested the performance and the accuracies improve from 28.6% to 90.2% and 20.6% to 92.0% respectively on the ModelNet40 dataset under the challenging z/SO3 mode.

4 Large-Scale Scene Segmentation on S3DIS

To evaluate the performance on semantic segmentation, we further conduct an experiment on S3DIS dataset (Armeni et al. 2016) which is a large-scale indoor scene dataset comprising 3D scans from Matterport scanners in 6 indoor areas including 271 rooms with each point is annotated with one of the semantic

Table 1: The details of our network. Here, K represents the number of categories, and N represents the number of input points.

Module	Output shape ($dims \times points$)
Classification / Retrieval	
Input tensor	3×1024
RISurConv	32×1024
RISurConv	64×512
RISurConv	128×256
RISurConv	256×128
RISurConv	512×1
Transformer Encoder	512×1
Fully connected	256×1
Fully connected	128×1
Softmax	$K \times 1$
Segmentation	
Input tensor	$3 \times N$
RISurConv	64×512
RISurConv	128×256
RISurConv	256×128
RISurConv	512×64
RISurConv	512×128
Skip connection	768×128
MLP	512×128
RISurConv	512×256
Skip connection	640×256
MLP	256×256
RISurConv	256×512
Skip connection	320×512
MLP	128×512
RISurConv	$K \times N$

labels from 13 categories. In this experiment, we use Area-5 for testing to better measure the generalization ability, and report the results in SO3/SO3 and z/SO3 scenario. Different from previous works that rotate the whole scan, we only rotate the instances such that it is closer to the real-life situation. We compare our method with traditional 3D deep learning methods such PointNet [2], PointCNN [1] and PointTransforer [7] as well as rotation invariant methods such as RIConv [5], GCAConv [4], and RIConv++ [6]. The results are shown in Table 2. From the results, we can see that our method outperforms RIConv++ by 2.1% mIoU, and works consistently for both rotation scenarios. Note that we use the original source codes of these methods and do not modify them by add normal computation. Thus, only RIConv++ and our method provide two versions for input with normal and without normal in this experiment.

Table 2: Comparisons of the semantic segmentation accuracy (mIoU, %) on the S3DIS dataset (Area-5).

Method	SO3/SO3	z/SO3
PointNet	39.1	31.2
PointCNN	51.2	30.6
PointTransformer	53.6	44.8
RIConv	53.8	53.8
GCAConv	54.3	54.1
RIConv++ (w/o normal)	57.8	57.8
RIConv++ (w/ normal)	58.0	58.0
Ours (w/o normal)	<u>59.3</u>	<u>59.3</u>
Ours (w/ normal)	**60.1**	**60.1**

5 Per-Class Accuracies

We further evaluate the accuracy of the classification task per object class under the z/SO3 mode. The per-class accuracies are shown in Table 3. Comparing to the results by non rotation invariant methods like PointNet [2], PointNet++ [3] and PointCNN [1], as well as rotation invariant methods like RIConv [5], GCAConv [4], and RIConv++ [6], we can see that our method outperforms other methods on most classes with a large margin. Our method achieves 1st (in bold) in 36 out of 40 classes, while RIConv [5], GCAConv [4], and RIConv++ [6] only have 2, 9, 9 classes achieving 1st, respectively.

References

1. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Adv. Neural Inform. Process. Syst. pp. 820–830 (2018)
2. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 652–660 (2017)
3. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Adv. Neural Inform. Process. Syst. pp. 5105–5114 (2017)
4. Zhang, Z., Hua, B.S., Chen, W., Tian, Y., Yeung, S.K.: Global context aware convolutions for 3d point cloud understanding. In: Int. Conf. 3D Vis. (2020)
5. Zhang, Z., Hua, B.S., Rosen, D.W., Yeung, S.K.: Rotation invariant convolutions for 3d point clouds deep learning. In: Int. Conf. 3D Vis. pp. 204–213 (2019)
6. Zhang, Z., Hua, B.S., Yeung, S.K.: Riconv++: Effective rotation invariant convolutions for 3d point clouds deep learning. Int. J. Comput. Vis. **130**(5), 1228–1243 (2022)
7. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Int. Conf. Comput. Vis. pp. 16259–16268 (2021)

Table 3: Per-class accuracy of object classification in z/SO3 scenario with the ModelNet40 dataset.

Network	aero	bathtub	bed	bench	bookshelf	bottle	bowl	car
PointNet [2]	12.0	2.0	8.0	10.0	15.0	14.0	5.0	12.0
PointNet++ [3]	53.0	2.0	18.0	10.0	29.0	22.0	20.0	13.0
PointCNN [1]	60.0	10.0	20.0	10.0	20.0	37.0	25.0	34.0
RIConv [5]	**100.0**	82.0	94.0	**80.0**	93.0	94.0	100.0	98.0
GCAConv [4]	**100.0**	90.0	98.0	**80.0**	95.0	97.0	**100.0**	98.0
RIConv++ [6]	**100.0**	90.0	97.0	**80.0**	**97.0**	94.0	**100.0**	**100.0**
Ours	**100.0**	**98.0**	**99.0**	**80.0**	**97.0**	**99.0**	95.0	**100.0**

	chair	cone	cup	curtain	desk	door	dresser	flower pot
PointNet [2]	9.0	15.0	0.0	0.0	16.3	5.0	8.1	0.0
PointNet++ [3]	32.0	20.0	15.0	45.0	2.3	30.0	9.3	15.0
PointCNN [1]	46.0	25.0	15.0	40.0	34.9	30.0	32.6	25.0
RIConv [5]	96.0	90.0	60.0	95.0	79.1	85.0	73.3	30.0
GCAConv [4]	**98.0**	90.0	55.0	95.0	81.4	80.0	68.6	10.0
RIConv++ [6]	97.0	95.0	70.0	**100.0**	83.7	85.0	72.1	15.0
Ours	**98.0**	**100.0**	**80.0**	**100.0**	**95.3**	**95.0**	**93.0**	**70.0**

	glass box	guitar	keyboard	lamp	laptop	mantel	monitor	night stand
PointNet [2]	4.0	36.0	5.0	15.0	15.0	4.0	11.0	3.5
PointNet++ [3]	11.0	47.0	50.0	10.0	15.0	10.0	36.0	1.2
PointCNN [1]	35.0	46.0	50.0	20.0	20.0	38.0	35.0	40.7
RIConv [5]	96.0	99.0	95.0	80.0	95.0	91.9	97.0	77.9
GCAConv [4]	**97.0**	**100.0**	95.0	**85.0**	**100.0**	93.0	98.0	73.3
RIConv++ [6]	**97.0**	100.0	**100.0**	95.0	100.0	94.0	96.0	80.2
Ours	96.0	**100.0**	**100.0**	**85.0**	**100.0**	**99.0**	**99.0**	**83.7**

	person	piano	plant	radio	range hood	sink	sofa	stairs
PointNet [2]	5.0	36.7	55.0	5.0	4.0	20.0	11.0	25.0
PointNet++ [3]	20.0	5.0	71.0	20.0	9.0	5.0	21.0	10.0
PointCNN [1]	15.0	34.0	26.0	10.0	28.0	20.0	32.0	30.0
RIConv [5]	85.0	90.8	83.0	55.0	87.0	75.0	92.0	85.0
GCAConv [4]	90.0	91.0	**93.0**	65.0	86.0	70.0	93.0	80.0
RIConv++ [6]	**100.0**	94.0	91.0	65.0	95.0	75.0	97.0	95.0
Ours	**100.0**	**97.0**	82.0	**95.0**	**96.0**	**85.0**	**98.0**	**95.0**

	stool	table	tent	toilet	tv stand	vase	wardrobe	xbox
PointNet [2]	5.0	3.0	5.0	20.0	4.0	26.3	0.0	10.0
PointNet++ [3]	10.0	9.0	15.0	13.0	2.0	85.0	15.0	20.0
PointCNN [1]	20.0	36.0	15.0	33.0	29.0	70.0	40.0	15.0
RIConv [5]	60.0	80.0	70.0	95.0	78.0	76.8	70.0	65.0
GCAConv [4]	75.0	84.0	95.0	99.0	81.0	77.0	**70.0**	75.0
RIConv++ [6]	85.0	81.0	95.0	99.0	87.0	84.0	45.0	90.0
Ours	**90.0**	**95.0**	**100.0**	**100.0**	**95.0**	**93.0**	55.0	**85.0**