DCPB: Deformable Convolution based on the Poincaré Ball for Top-view Fisheye Cameras

Xuan Wei
School of Automation, Southeast University
Nanjing 210096, China
wx1204@seu.edu.cn

Zhidan Ran
School of Automation, Southeast University
Nanjing 210096, China
230218198@seu.edu.cn

Xiaobo Lu
School of Automation, Southeast University
Nanjing 210096, China
xblu2013@126.com

Abstract

The accuracy of the visual tasks for top-view fisheye cameras is limited by the Euclidean geometry for pose-distorted objects in images. In this paper, we demonstrate the analogy between the fisheye model and the Poincaré ball and that learning the shape of convolution kernels in the Poincaré Ball can alleviate the spatial distortion problem. In particular, we propose the Deformable Convolution based on the Poincaré Ball, named DCPB, which conducts the Graph Convolutional Network (GCN) in the Poincaré ball and calculates the geodesic distances to Poincaré hyperplanes as the offsets and modulation scalars of the modulated deformable convolution. Besides, we explore an appropriate network structure in the baseline with the DCPB. The DCPB markedly improves the neural network's performance. Experimental results on the public dataset THEODORE show that DCPB obtains a higher accuracy, and its efficiency demonstrates the potential for using temporal information in fisheye videos.

1. Introduction

Visual tasks for top-view fisheye cameras are challenging but with valuable practical significance in real-world scenarios [18,45]. Due to their large field of view, top-view fisheye cameras are the most cost-effective devices to capture 360○ content and cover a more comprehensive range of applications in visual surveillance [17]. A traditional perspective camera samples a field of view of the 3D scene projected onto a 2D plane with relative positions in the real world [35]. In contrast, a top-view fisheye camera captures the entire view surrounding its optical center. Thus, top-view fisheye cameras provide more spatial information than traditional perspective cameras. However, the geometric transformations inherent in fisheye cameras cause spatial distortion, leading to magnifying objects near the center of fisheye images while the objects away from the center of fisheye images shrink [41,46]. The spatial distortion considerably increases the difficulty of accurate visual tasks for top-view fisheye cameras, demanding the algorithms to be robust. Besides, the algorithms trained on perspective cameras usually perform poorly on top-view fisheye cameras [49,51]. To solve these problems, many researchers are committed to adapting Convolutional Neural Networks (CNNs) to the severe distortion of fisheye images for the higher accuracy of visual tasks.

In object detection, [17,24] proposed CNNs that predict arbitrarily rotated bounding boxes in a fisheye image to increase IoU. [10,36] rotated each fisheye image and applied YOLO [31] only to the upper center part of the image, where people usually appear upright. These methods only deal with the input and output of CNNs and ignore that a convolution kernel may not be appropriate for the distorted object. Deformable convolution [14] is proposed for addressing the issue that geometric variations due to scale, pose, viewpoint, and part deformation degrade the performance of CNNs in object recognition and detection. In deformable convolution, the grid sampling locations of standard convolution are each offset by displacements learned for the preceding feature maps. Deformable ConvNets v2 (DCNv2) [50] introduced modulated deformable convolu-

*This work is supported by the National Natural Science Foundation of China under grant 62271143, the National Key R&D Program of China under Grant 2020YFB1600700, and the Big Data Center of Southeast University.

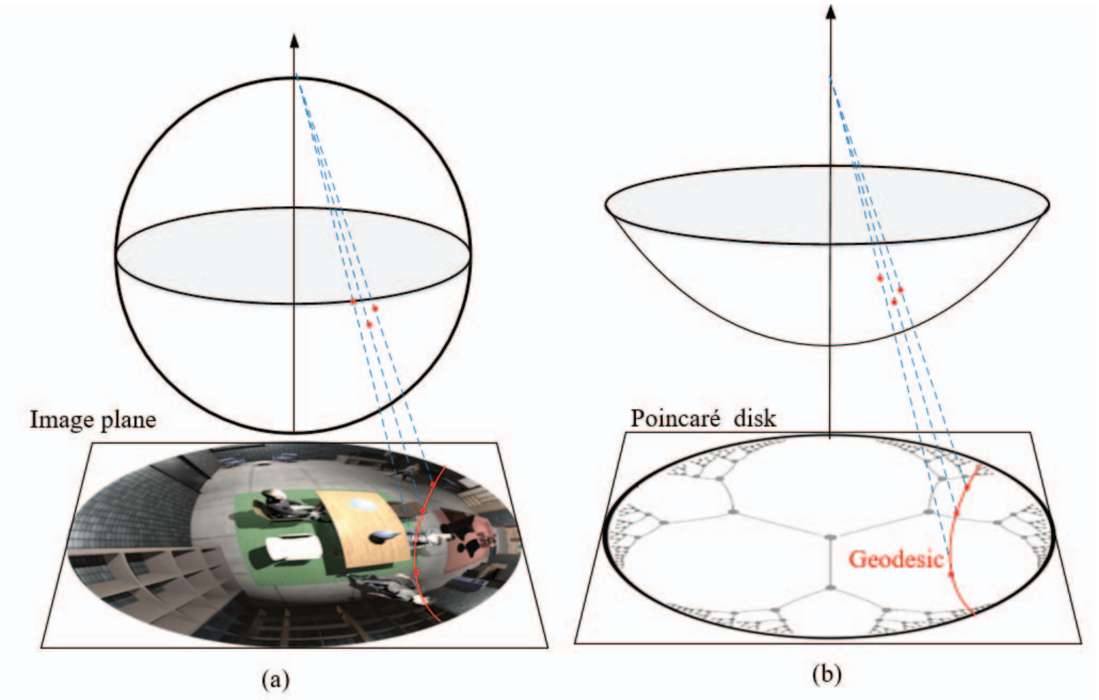

Figure 1. (a) The projection model of the top-view fisheye camera. (b) The projection model of hyperboloid on the Poincaré disk.

tion to strengthen the ability of the model to vary the spatial distribution. [43] proposes a deformable subnetwork that can generate 4-dimensional deformation coefficients and perform part alignment to handle object deformation. [4] replaces the fixed convolution layer and pooling layer in Cascade-RCNN [6] with the deformable convolution layer and deformable pooling layer to conduct object detection in top-view fisheye cameras. To eliminate 2D-to-Sphere intrinsic sampling distortions of fisheye images, [7] uses deformable convolutions to extract omnidirectional features with non-deformable Receptive Fields. These methods build deformable kernel functions on the convolution layers in the Euclidean space and ignore that top-view fisheye images have similar geometric properties with hyperbolic space [1]. To solve the problem that the neural network in Euclidean space lacks the ability to express the features of fisheye images, we try to find an appropriate geometric model in hyperbolic space and learn the features in it.

The Poincaré ball, an n-dimensional hyperbolic geometric model, is a stereographic projection of the hyperbolic space. As its two-dimensional form, the Poincaré disk can achieve a more accurate approximation of the projection model of the top-view fisheye camera, as shown in Fig. 1. The Poincaré ball is a conformal mapping that preserves angles between distorted lines [1]. Thus, the geodesic in the Poincaré disk corresponds to any arc perpendicular to the disk's boundary or diameter. By considering the surface area of a hypersphere of increasing radius centered at a particular point, the Poincaré ball and fisheye images can be seen to grow exponentially. In addition, the objects near the center of fisheye images are magnified while the objects away from the center shrink, like the induced distance [40] in the Poincaré ball. Therefore, we think learning the relevant features of fisheye images in the Poincaré ball is feasible.

In this paper, to improve the ability of the convolution kernel to extract distorted features from top-view fisheye images, we propose the Deformable Convolution in the Poincaré Ball (DCPB) for top-view fisheye cameras. We embed the features of the fisheye image in the Euclidean space into the Poincaré ball and then obtain the offsets and modulation scalars of the modulated deformable convolution through the Graph Convolution Network (GCN) in the hyperbolic space. To thoroughly verify the increased modeling capacity of the DCPB, we conduct experiments on image semantic segmentation with the synthetic segmentation dataset THEODORE [34]. Specially, we incorporate the DCPB into the segmentation networks UNet [32], and we show that our method improves the performance of CNN semantic segmentation on synthetic distortions.

Our contributions can be summarized as follows:

- We propose the DCPB which is a novel convolution method to learn the shape of the kernel in the Poincaré ball for top-view fisheye cameras, enabling the CNNs adapted to the severe distortion for the higher accuracy of visual tasks.

- We propose the method to project the feature from the Poincaré ball back to the Euclidean space by calculating the Geodesic distance from features to Poincaré hyperplanes, which enhances the information learning of the CNNs between different spaces.

- We explore an appropriate network structure for the DCPB in the segmentation networks UNet and verify the effectiveness of the convolution methods on the synthetic top-view fisheye segmentation dataset.

2. Related Works

2.1. Convolution kernel adaptation

To adapt to rotational images, [12, 26, 42] proposed spherical convolution and convolutions in group space. [11, 13] developed efficient convolution algorithms for spherical signals and created building blocks that satisfy a generalized Fourier theorem to detect patterns independently from their location on the sphere. [50] proposed modulated deformable convolution to deform convolution kernel and enhanced the adaptability and computational efficiency of the convolution. To address the spatial distortion problem of fisheye cameras, [16] proposed Restricted Deformable Convolution (RDC) for pixel-wise prediction tasks in fisheye images. [30] used deformable convolution to adapt standard CNN models on fisheye images to capture non-linear transformations. [41] developed a novel convolution method named the Rotation Mask Deformable Convolution (RMDC), which rotates convolution kernels and introduces the Center-Fixed Deformable Convolution.

2.2. Neural Networks on Hyperbolic spaces

Hyperbolic space can be thought of as a continuous version of trees, and as such, it is naturally equipped with hierarchical model structures [28]. Recent research has proven that many types of complex data (e.g., graph data) from many fields exhibit a highly non-Euclidean latent characteristic [5]. In representation learning existing approach to embedding hierarchical multi-relational graph data in hyperbolic space has outperformed Euclidean models [3, 37]. In machine learning, hyperbolic representations recently significantly outperformed Euclidean embeddings for hierarchical, taxonomic, or entailment data [20, 33]. In computer vision, [1] introduced FisheyeHDK, a hybrid neural network that combines hyperbolic and Euclidean convolution layers to learn the shape and weights of deformable kernels.

3. Methods

3.1. Background and preliminaries

Hyperbolic geometry of the Poincaré ball. The hyperbolic space has five isometric models, and we work in Poincaré ball like [21, 25]. The Poincaré ball $\mathbb{B}_c^d$ of radius $1/\sqrt{c}, c > 0$ corresponds to a d-dimensional Riemannian manifold, where $\mathbb{B}_c^d = \left\{x \in \mathbb{R}^d \mid \|x\| < 1/\sqrt{c}\right\}$ is an open ball. Its metric tensor is given by:

$$g_x^{\mathbb{B}} = \lambda_x^2 g_x^{\mathbb{E}}, \lambda_x := \frac{2}{1-\|x\|^2}. \tag{1}$$

$g_x^{\mathbb{E}} = I^n$ denotes the Euclidean metric tensor and $\|\cdot\|$ denotes the Euclidean norm. Thus, the hyperbolic metric tensor is conformal to the Euclidean one. As Fig. 2 shows, the summation of two points x, y in the Poincaré ball is defined by $M\ddot{o}bius\ addition$ [38]:

$$x \oplus_c y := \frac{\left(1+2c\langle x,y\rangle + c\|y\|^2\right)x + \left(1-c\|x\|^2\right)y}{1+2c\langle x,y\rangle + c^2\|x\|^2\|y\|^2}. \tag{2}$$

The induced distance which is measured along a geodesic (i.e., the shortest path between the points $x, y \in \mathbb{B}_c^d$) is given by:

$$d_{\mathbb{B}_c^d}(x,y) = \frac{1}{\sqrt{c}}\cosh^{-1}\left(1+2\frac{\|x-y\|^2}{\left(1-c\|x\|^2\right)\left(1-c\|y\|^2\right)}\right). \tag{3}$$

Because the effect of a neural network is poor if it is directly applied to hyperbolic space, a common approach is to project the point x in hyperbolic space onto its tangent space $\mathcal{T}_x\mathbb{B}_c^d$ which is a d-dimensional Euclidean space, and apply a neural network to the tangent space [21].The exponential map $exp_x^c : \mathcal{T}_x\mathbb{B}_c^d \to \mathbb{B}_c^d$ allows one to move on the manifold from x in the direction of a vector $v \in \mathcal{T}_x\mathbb{B}_c^d$, tangential to $\mathbb{B}_c^d$ at x. And the inverse is the logarithmic map $log_x^c : \mathbb{B}_c^d \to \mathcal{T}_x\mathbb{B}_c^d$. For the Poincaré ball and $v \neq 0, x \neq y$, these are defined as:

$$\exp_x^c(v) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right), \tag{4}$$

$$\log_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c}\tanh^{-1}\left(\sqrt{c}\,\|-x \oplus_c y\|\right)\frac{-x \oplus_c y}{\|-x \oplus_c y\|}. \tag{5}$$

Using the more straightforward form when $x = 0$, [21] shows that the linear mapping $M : \mathbb{R}^n \to \mathbb{R}^m$ can be applied on the Poincaré ball by projecting a point $x \in \mathbb{B}_c^d$ onto the tangent space at $0 \in \mathbb{B}_c^d$ with the logarithmic map $\log_0^c(x)$, performing matrix multiplication in the Euclidean tangent space $\mathcal{T}_x\mathbb{B}_c^d$, and finally projecting back to $\mathbb{B}_c^d$ by the exponential map $\exp_0^c(x)$, i.e.:

$$f_{out}(x) := \exp_{\mathbf{0}}^c\left(M\left(\log_{\mathbf{0}}^c(x)\right)\right). \tag{6}$$

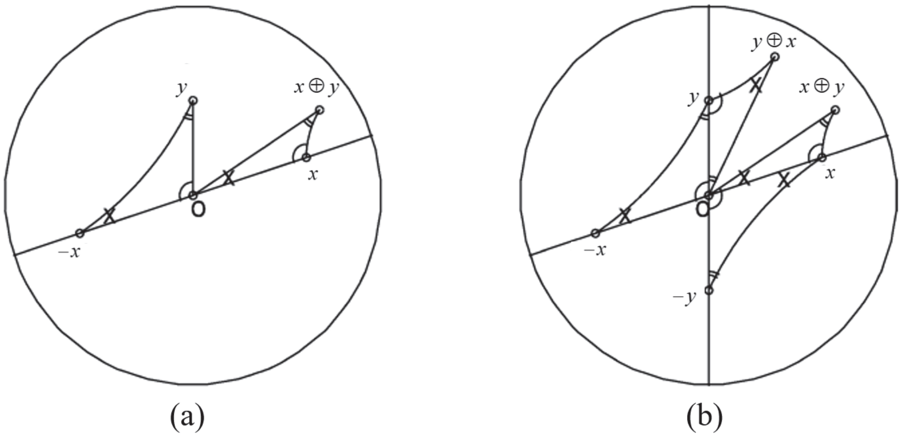

Figure 2. (a) The point $x \oplus y$ is the point obtained by replacing the triangle $\triangle(-x, o, y)$ parallel over the line passing through the points $-x, x$ to a triangle $\triangle(o, x, x \oplus b)$. (b) the geometrical description of the summation $y \oplus x$ [39].

Modulated deformable convolution. Deformable convolution is introduced to extend the regular grid sampling locations used in convolutions with 2D offsets, which are learned from the input feature maps of the previous layer. Given a convolutional kernel of K grids sampling, w_k and p_k denote the weight and pre-specified offset for the k-th location, respectively. For a 3×3 convolutional kernel of dilation 1, $K = 9$ and $p_k \in \{(-1,-1), (-1,0), ..., (1,0), (1,1)\}$. Let $f(x)$ denote the input feature value at a location x, and the output feature $f_{out}(x)$ value of the deformable convolution can be calculated as:

$$f_{out}(x) = \sum_{k=1}^{K} w_k \cdot f(x + p_k + \Delta p_k), \tag{7}$$

where Δp_k is the learnable offset for the k-th location in grids. Bilinear interpolation is applied for computing $f(x + p_k + \Delta p_k)$ because the location $x + p_k + \Delta p_k$ is fractional. To further strengthen the capability of the deformable convolution in manipulating spatial support regions, DCNv2 [50] introduced a modulation mechanism as:

$$f_{out}(x) = \sum_{k=1}^{K} w_k \cdot f(x + p_k + \Delta p_k) \cdot \Delta m_k, \tag{8}$$

where Δm_k is the learnable modulation scalar which lies in the range [0, 1].

Both the offset Δp_k and the modulation scalar Δm_k are learned from the input feature map x of the previous layer via a separate convolution layer. The convolution results of Δm_k are activated by a sigmoid function. The shape of the output of the offset convolution layer is like the input feature map, and the difference is that the dimension of the channel is $2K$, including offset values in the vertical and horizontal direction for K grid. Similarly, the output of the modulation scalar convolution layer has K channels. As such, for each grid sampling in the feature map, we need to learn $3K$ parameters, and the following shows how to obtain them in the Poincaré ball.

3.2. Deformable Convolution based on Poincaré Ball

DCPB Architecture. Fig. 3 demonstrates the architecture of the Deformable Convolution based on the Poincaré Ball (DCPB). We embed the input feature into an 8-connected graph and map it to a Poincaré Ball. To implement the feature updates of the graph, the GCN is performed on the Poincaré Ball to transform and aggregate the features. Then we construct learnable Poincaré hyperplanes and compute geodesic distances from the feature of each node to Poincaré hyperplanes as the parameters in the deformable convolution. The proposed architecture is described minutely in the following.

Graph embedding. In a CNN, the convolution layer can aggregate the feature information in the receptive field in the Euclidean space. Still, it is difficult to apply to the features in the Poincaré ball. Therefore, a feasible method is to embed the feature in the Euclidean space into a graph in the Poincaré ball and use a Graph Convolution Network (GCN) to aggregate features of adjacent nodes. A feature map can be regarded as rectangular grids with channels and represented as arrays (e.g., $512 * 512 * 3$), whereas we can also think of it as a graph with a regular structure, where each pixel represents a node and is connected via an edge to adjacent pixels. Similar to 3×3 convolution, we embed a feature map with N_{in} channels into a graph where a non-border node has eight neighbors. The information stored at each node is a N_{in}-dimensional vector representing the channel values of the pixel in the feature map. Fig. 4 (a) and (b) demonstrate embedding a 5×5 feature map to an 8-

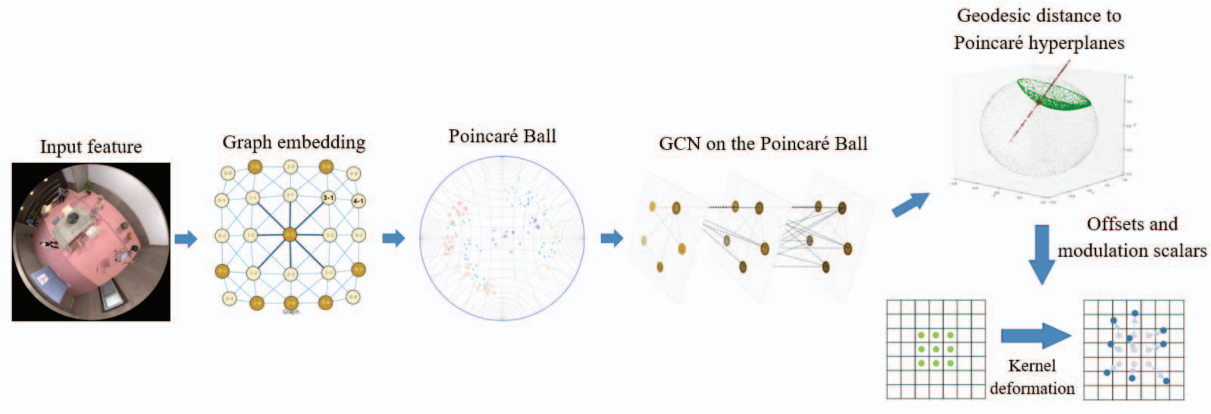

Figure 3. The architecture of the DCPB.

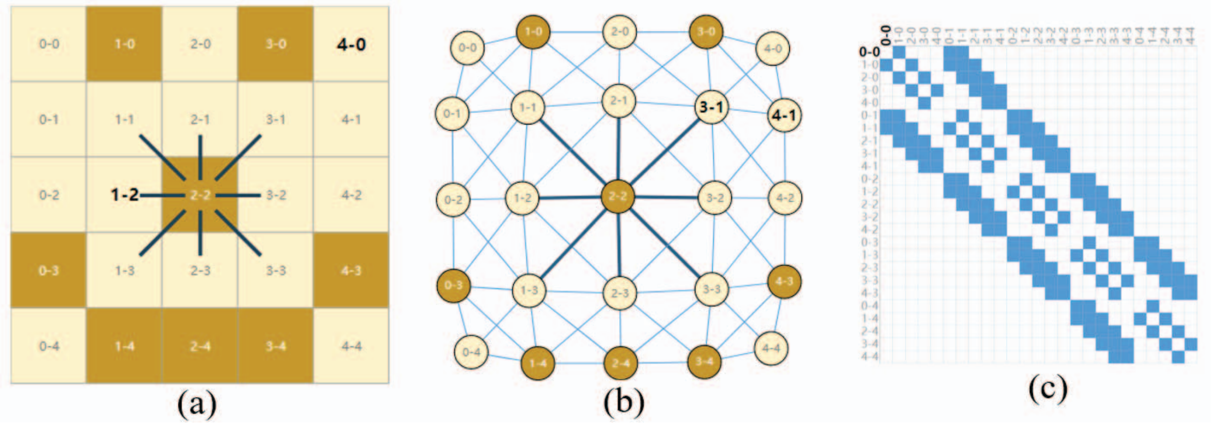

Figure 4. (a) A 5 × 5 feature map. (b) An 8-connected graph. (c) The adjacency matrix. Note that these three representations are different views of the same piece of data.

connected graph. As Fig. 4 (c) shows, the adjacency matrix A preserves explicit relationships between nodes and needs to be calculated for the GCN.

GCN in the Poincaré Ball. After graph embedding and computing the adjacency matrix A, we can use the GCN to perform message passing between features in the Poincaré Ball. We base our work on the GCN proposed in [23], which introduces a first-order approximation of ChebNet [15]. The GCN encapsulates each node's hidden representation by aggregating feature information from its neighbors. After feature aggregation, a nonlinear transformation is applied to the resulting outputs. As such, the message from node y to its receiving the neighbor node x is computed as:

$$m_y = W\tilde{A}_{xy}f(x), \tag{9}$$

where $f(x)$ is the feature with N_{in} channels at the node x, $W \in \mathbb{R}^{N_{in}} \to \mathbb{R}^{N_{mid}}$ is the weight matrix from input to mid layer, and $\tilde{A} = I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ captures the connectivity of the graph. Here D is the diagonal degree matrix of the graph:$D_{ii} = \sum_j (A_{ij} + I_{ij})$, and $\tilde{A}$ is computed by adding the identity matrix I to the normalized adjacency matrix $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. By summing up all the messages from its neighbors and applying the activation function , we can obtain the information propagates (i.e., the output features) of the graph:

$$f_{out}(x) = \sigma\left(\sum_{y\in N(x)} m_y\right) = \sigma\left(\sum_{y\in N(x)} W\tilde{A}_{xy}f(x)\right), \tag{10}$$

where $N(x)$ is the neighborhood of x, and $y \in N(x)$ has an edge pointing to x.

For the GCN in the Poincaré Ball, we map the input feature in a Poincaré Ball $\mathbb{B}_1^{N_{in}}$ of radius 1 by clamping the value of the input feature to [0, 1], and the dimension N_{in} is equal to the input feature. As equation 6 shows, linear mapping can be applied to the tangent space of the Poincaré ball. Similarly, the GCN in the Poincaré Ball is conducted by projecting the input feature $f(x) \in \mathbb{B}_1^{N_{in}}$ onto the tangent space at $0 \in \mathbb{B}_1^{N_{in}}$ with the logarithmic map, performing the GCN in the Euclidean tangent space $\mathcal{T}_{f(x)}\mathbb{B}_1^{N_{in}}$, and finally projecting back to $\mathbb{B}_1^{N_{mid}}$ by the exponential map. Note that the dimensions of GCN input and output are different; thus, the output is projected back to another Poincaré ball $\mathbb{B}_1^{N_{mid}}$.As such, the information propagates of the GCN in the Poincaré ball is calculated as:

$$f_{out}(x) = \sigma\left(exp_0^1\left(\sum_{y\in N(x)} W\tilde{A}_{xy}\left(log_0^1 f(x)\right)\right)\right). \tag{11}$$

Geodesic distance from features to Poincaré hyperplanes. Because of the analogy between the fisheye model and the Poincaré ball, we can directly map the features in the Euclidean space to the Poincaré Ball. However, the offset and modulation parameters in the deformable convolution are very different from the data in Poincaré Ball, and the output feature of the GCN in the Poincaré Ball cannot be used as these parameters directly. Therefore, we construct N_{out} learnable Poincaré hyperplanes and compute N_{out} geodesic distances from the feature of each node to Poincaré hyperplanes as the parameters in the deformable convolution. Here, N_{out} is equal to $3K$, the number of the parameters to learn for each grid sampling in the feature map, and the geodesic distance is the data in the Euclidean space.

For $p \in \mathbb{B}_1^{N_{mid}}, a \in \mathcal{T}_p\mathbb{B}_1^{N_{mid}} \setminus 0$, a Poincaré hyperplane can be defined as:

$$\begin{aligned}\tilde{H}^1_{a,p} &:= \left\{x \in \mathbb{B}_1^{N_{mid}} : \left\langle \log_p^1(x), a\right\rangle_p = 0\right\} \\ &= \left\{x \in \mathbb{B}_1^{N_{mid}} : \langle -p \oplus x, a\rangle = 0\right\},\end{aligned} \tag{12}$$

where $\tilde{H}^1_{a,p}$ can also be described as the union of images of all geodesics in $\mathbb{B}_1^{N_{mid}}$ orthogonal to a and containing p. Then we can calculate the geodesic distance from a feature in the Poincaré Ball to the hyperplane as:

$$\begin{aligned} d\left(x, \tilde{H}_{a,p}\right) &:= \inf_{w \in \tilde{H}_{a,p}} d(x,w) \\ &= \sinh^{-1}\left(\frac{2\left|\langle -p \oplus x, a\rangle\right|}{\left(1-\|-p\oplus x\|^2\right)\|a\|}\right).\end{aligned} \tag{13}$$

3.3. Other details

The geodesic distances of each node have $3K$ channels, where the first $2K$ channels correspond to the learned offsets Δp_k, and a sigmoid function activates the remaining K channels to obtain the modulation scalars Δm_k. Δp_k and Δm_k are initialized to 0 and 0.5, respectively. The learning rates of the added GCN layer and the Poincaré hyperplane are set to 0.1 times those of the other standard convolution layers. The channel N_{mid} of the output of the GCN layer is the same as the channel N_{out}, and we select the Rectified Linear Unit (ReLU) as the activation function in the GCN in the Poincaré ball.

In a CNN, features from the lower convolution layers encode low-level spatial visual information like edges, corners, circles, etc. Features from the higher convolution layers encode high-level semantic information and weak spatial information, including object- or category-level evidence. For high-level convolution kernels, the extraction of semantic information and integration of spatial information from the lower convolution layers are improved by the DCPB. However, for low-level convolution kernels, variety ensures their ability to extract spatial details information, and the DCPB will fluctuate their spatial structures. We only apply the DCPB to replace the standard 3×3 convolutions in higher convolution layers. For example, in a standard UNet, the end of the encoder and decoder includes higher convolution layers with more channels of convolution kernels, and the DCPB is adopted at these layers.

4. Experiments

4.1. Experimental settings

Currently, top-view fisheye datasets are scarce, and THEODORE [34] is the unique segmentation dataset, a synthetic downside indoor scenes dataset containing 100k images with 16 classes. Our segmentation experiment is carried out on 10k images of THEODORE. All segmentation models are trained with the same configuration. We use an adaptive learning rate momentum algorithm (Adam) with an initial learning rate of 0.0001, a momentum of 0.9, and a weight decay of 0.0001. Random horizontal flip is used for data augmentation. The input resolution of networks is 512×512, and the batch size is 32. Besides, we train models for 40000 iterations. We adopt the mean of class-wise intersection over union (mIoU) and pixel accuracy (mPA) as the evaluation metrics for segmentation evaluation and select the standard cross entropy as the loss function. In all experiments, 80% of the images are used for training and validation and 20% for testing. We perform all experiments under Pytorch 1.9, CUDA 11.1, and CUDNN 7.6.5 on four NVIDIA RTX 2080Ti GPUs.

4.2. Ablation Study

We first evaluate our proposed method in the baseline model UNet. Standard UNet has 8 stages, in which the first 4 stages (Down1$\sim$Down4) are used as encoders to downsample and extract features, and the decoders include the last 4 stages (UP1$\sim$UP4) to reconstruct the segmentation images. Here experiments are conducted using variants of our method in the UP1 stage of the UNet to evaluate their semantic segmentation performance. Table 1 shows the ef-

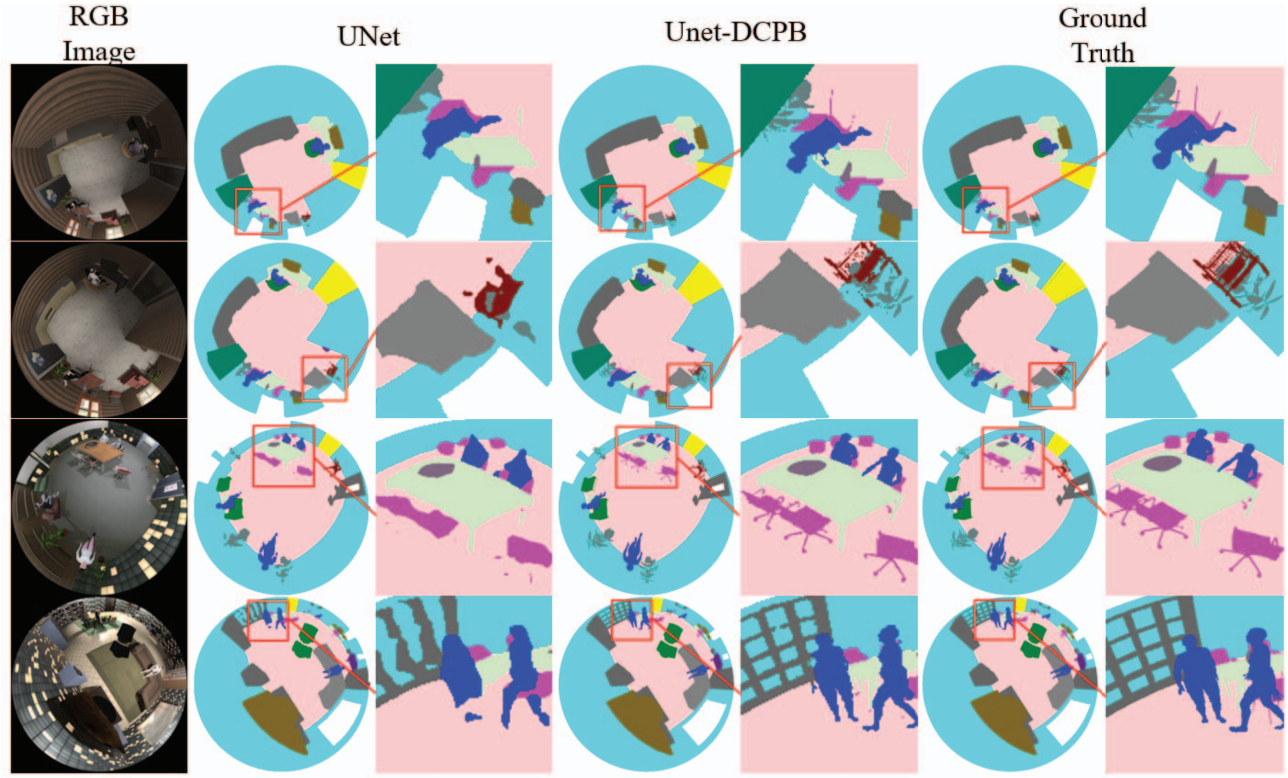

Figure 5. Qualitative comparison of the semantic segmentation results on images from THEODORE dataset. Columns show the segmentation results from baseline with different convolution methods.

Table 1. Ablation study of enriched deformable convolution on baseline model. In the setting column, “DC” stands for modulated deformable convolution. “GCNE” and “GCNP” mean that the offsets and modulation scalars of the modulated deformable convolution are obtained by graph convolution network (GCN) in the Euclidean space and Poincaré ball, respectively. Also, “GD” stands for computing the geodesic distances to Poincaré hyperplanes.

Method	Setting	mIoU (%)	mPA(%)	FPS
Baseline	Regular	90.85	94.54	**13.8**
	DC [50]	91.97	94.96	13.7
Enriched deformation	DC+GCNE	92.57	95.58	13.7
	DC+GCNP	92.74	95.72	13.5
	DC+GCNP+GD(DCPB)	**93.05**	**96.12**	13.4

fects of enriched deformation methods from ablation experiments. The inference speed (FPS) is estimated from serial images of testing data using an Nvidia RTX 2080Ti GPU.

The baseline with regular CNN modules obtains an mIoU score of 90.85% and an mPA score of 94.54% for UNet. By replacing regular CNN modules with DC, the accuracy of the UNet steadily improves, with gains between 1.12% and 0.42% for the mIoU and mPA scores. Computing the offsets and modulation scalars of DC in GCN increases mIoU and mPA scores by 0.6% and 0.62%, respectively. By upgrading the Euclidean space to a Poincaré ball, mIoU and mPA scores steadily improve by 0.17% and 0.14%. When we compute the geodesic distances from the feature of each node to Poincaré hyperplanes as the parameters in the deformable convolution, we obtain further gains between 0.31% and 0.4% in mIoU and mPA scores. In total, the DCPB method yields 93.05% mIoU and 96.12% mPA scores on UNet. Note that although the inference speeds of models are harmed slightly, the models with DCPB can still meet the real-time requirements.

Fig. 5 shows another few parsing results on the test sets and demonstrates the visual comparison of the DCPB. With DCPB, the model pays more attention to distorted objects in the edge region and contains more accurate and detailed structures than the baseline.

To explore an appropriate network structure with DCPB,

Table 2. Per-class results on THEODORE test set. We choose ResNet-18 pre-trained on MS-COCO as the backbone of DeepLabV3, DeepLabV3+, PSPNet, BiSeNetV2, DANet, and DUNet. Other methods are trained only on the THEODORE training set. The comparison experiments are conducted on THEODORE test set

Method	Pers.	Door	Chair	Wall	Floor	Table	Sofa	Furn.	Lamp	Deco.	Plant	Scre.	Bed	Frid.	Whee.	Armch.	FPS	mIoU
ESPNetV2 [27]	59.5	88.2	44.2	93.8	90.6	69.6	84.7	83.1	55.8	29.3	72.2	81.9	83.9	87.3	31.2	74.5	12.2	70.6
CGNet [44]	61.9	91.1	52.0	94.5	91.7	75.3	87.6	85.0	65.2	46.0	63.7	84.2	86.0	89.0	37.8	78.5	15.1	74.3
ENet [29]	64.5	93.6	53.9	95.5	92.8	78.5	89.8	88.3	46.9	44.9	68.2	86.9	88.8	91.6	37.4	80.7	13.7	75.1
DeepLabV3 [8]	76.5	95.7	70.0	96.3	94.3	88.3	94.8	92.6	74.3	78.7	60.2	91.3	93.3	94.8	50.2	88.3	**15.6**	83.7
PSPNet [48]	77.8	96.9	75.0	97.2	95.8	90.0	95.5	93.8	81.2	78.0	66.2	93.2	94.8	95.2	49.8	90.6	14.9	85.7
BiSeNetV2 [47]	79.7	97.1	74.9	97.3	95.9	89.9	95.5	94.1	81.8	77.2	67.5	93.3	95.0	95.7	51.6	90.5	13.8	86.1
DANet [19]	79.4	97.2	76.3	97.5	96.1	90.6	95.8	94.2	82.7	79.0	67.8	93.7	95.4	96.0	49.9	91.2	13.9	86.4
DeepLabV3+ [9]	79.3	95.5	76.9	97.3	96.4	90.2	93.7	93.5	86.0	77.6	76.6	93.0	93.8	94.9	58.0	91.0	15.4	87.1
SegNet [2]	83.0	96.2	80.2	97.6	96.4	91.7	94.6	95.1	87.2	81.4	80.7	94.5	94.5	96.0	61.4	92.1	15.2	88.9
DUNet [22]	86.0	97.3	81.7	98.1	97.2	93.0	96.7	95.8	86.7	83.5	78.9	95.1	96.2	96.3	62.7	93.4	13.7	89.9
UNet [32]	89.0	97.5	83.8	98.0	97.4	93.3	96.5	94.4	90.0	85.0	83.4	95.2	94.8	93.3	68.1	93.9	13.8	90.8
UNet-DCPB	**91.4**	**98.4**	**87.6**	**98.7**	**98.2**	**95.3**	**97.6**	**97.3**	**91.7**	**88.8**	**84.6**	**96.9**	**97.4**	**97.7**	**71.8**	**95.5**	13.4	**93.1**

we use DCPB to replace the 3 × 3 standard convolution module in different baseline stages. As reported in Table 3, DCPB at various stages of UNet can improve the segmentation performance, and substituting it for standard CNN modules in the UP1 stage achieves the optimal results in mIoU score and FPS. As section 3.3 states, deformable convolution improves the extraction of semantic information for high-level convolution kernels but may fluctuate the spatial structures of low-level convolution kernels. Therefore, it is a better choice to replace the layer which contains high-level semantic information such as stages Down4 and UP1 with DCPB.

Table 3. Performance comparison of UNet with DCPB in the different stages.

Stage	mIoU (%)	mPA(%)	FPS
Down3	92.99	95.88	12.9
Down4	93.02	**96.15**	13.3
Up1	**93.05**	96.12	**13.4**
Up2	92.78	95.92	13.2
Up3	92.69	96.01	12.6

4.3. Comparison experiments

With deformable convolution methods. In order to prove the superiority of DCPB in visual tasks for downside fisheye images, we incorporate the Restricted Deformable Convolution [16] and Rotation-Mask Deformable Convolution [41] into UNet to investigate the effect of other deformable convolution methods. Besides, the hyperbolic deformable convolution in the FisheyeHDK [1] is applied in UNet. The FisheyeHDK also obtains the offsets in deformable convolution in the Poincaré ball. The difference is that FisheyeHDK maps the features to the tangent space of the Poincaré ball and does not compute geodesic distances to Poincaré hyperplanes. The experimental data are presented in Table 4. It can be seen from the data that DCPB performs better than the previous ones on the THEODORE, suggesting that our method is helpful for visual tasks in top-view fisheye images. In terms of efficiency, there is also not much disparity between our and other methods.

Table 4. Performance comparison of DCPB and other deformable convolution methods. "DC" stands for modulated deformable convolution. "RDC" and "RMDC" stand for Restricted Deformable Convolution and Rotation-Mask Deformable Convolution, respectively. "DSN" stands for deformable convolution which generates 4-dimensional deformation coefficients. Results are reported on the THEODORE test set.

Method	mIoU (%)	mPA(%)	FPS
DC [50]	91.97	94.96	**13.7**
RDC [16]	92.36	95.22	13.7
DSN [43]	92.55	95.37	13.6
RMDC [41]	92.89	95.9	13.5
FisheyeHDK [1]	92.07	95.48	13.5
DCPB	**93.05**	**96.12**	13.4

With common methods. As a final experiment on DCPB, we compare the performance of our algorithm with other common methods in semantic segmentation in top-view fisheye images. As Table 2 demonstrates, it is apparent that UNet with DCPB outperforms other methods with a notable advantage-the highest accuracy in all 16 classes, further confirming the superior capability of DCPB.

5. Conclusion

This paper analyzes the analogy between the fisheye model and the Poincaré ball to inspire future research on hyperbolic geometry for learning deformations from top-view fisheye images. Specially, we present the Deformable

Convolution based on the Poincaré Ball against complex situations in top-view fisheye images. The DCPB establishes a novel approach that learns the parameters of the modulated deformable convolution in the Poincaré ball and outperforms the convolution methods with corresponding Euclidean architectures on sequential data with an implicit hierarchical structure. We implement the DCPB by conducting GCN in the Poincaré ball and computing the geodesic distances to Poincaré hyperplanes as the offsets and modulation scalars of the modulated deformable convolution. To evaluate the effectiveness of our method, we perform experiments on semantic segmentation with the public dataset THEODORE. Besides, we also explore an appropriate network structure with the DCPB. And the DCPB has the potential to be valid for analyzing data from more visual tasks for top-view fisheye cameras.

References

[1] Ola Ahmad and Freddy Lecue. Fisheyehdk: Hyperbolic deformable kernel learning for ultra-wide field-of-view image recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):5968–5975, 2022.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] Jun Bao and Hongzhe Liu. Object detection in fisheye images combining deformable convolutional networks. *Computer Engineering*, 41:248–255, 2021.

[5] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[7] Xiongli Chai, Feng Shao, Qiuping Jiang, Xiangchao Meng, and Yo-Sung Ho. Monocular and binocular interactions oriented deformable convolutional networks for blind quality assessment of stereoscopic omnidirectional images. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3407–3421, 2021.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. pages 801–818, 2018.

[10] Sheng-Ho Chiang, Tsaipei Wang, and Yi-Fu Chen. Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches. *Image and Vision Computing*, 105:104069, 2021.

[11] Taco Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Convolutional networks for spherical signals. *arXiv preprint arXiv:1709.04893*, 2017.

[12] Taco Cohen and Max Welling. Group equivariant convolutional networks. *International conference on machine learning*, pages 2990–2999, 2016.

[13] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[14] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[16] Liuyuan Deng, Ming Yang, Hao Li, Tianyi Li, Bing Hu, and Chunxiang Wang. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Transactions on Intelligent Transportation Systems*, 21(10):4350–4362, 2020.

[17] Zhihao Duan, Ozan Tezcan, Hayato Nakamura, Prakash Ishwar, and Janusz Konrad. Rapid: rotation-aware people detection in overhead fisheye images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 636–637, 2020.

[18] Andrea Eichenseer, Michel Bätz, and André Kaup. Motion estimation for fisheye video with an application to temporal resolution enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8):2376–2390, 2019.

[19] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.

[20] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. pages 1646–1655, 2018.

[21] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

[22] Qiangguo Jin, Zhaopeng Meng, Tuan D Pham, Qi Chen, Leyi Wei, and Ran Su. Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*, 178:149–162, 2019.

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[24] Shengye Li, M. Ozan Tezcan, Prakash Ishwar, and Janusz Konrad. Supervised people counting using an overhead fisheye camera. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2019.

[25] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[26] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. *IEEE International Conference on Computer Vision (ICCV)*, pages 5058–5067, 2017.

[27] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9190–9200, 2019.

[28] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[29] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[30] Clément Playout, Ola Ahmad, Freddy Lecue, and Farida Cheriet. Adaptable deformable convolutions for semantic segmentation of fisheye images in autonomous driving systems. *arXiv preprint arXiv:2102.10191*, 2021.

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, pages 234–241, 2015.

[33] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. *International conference on machine learning*, pages 4460–4469, 2018.

[34] Tobias Scheck, Roman Seidel, and Gangolf Hirtz. Learning from theodore: A synthetic omnidirectional top-view indoor dataset for deep transfer learning. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 932–941, 2020.

[35] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360-degree imagery. *Advances in Neural Information Processing Systems*, 30, 2017.

[36] Masato Tamura, Shota Horiguchi, and Tomokazu Murakami. Omnidirectional pedestrian detection by rotation invariant training. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1989–1998, 2019.

[37] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

[38] Abraham A Ungar. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2):135–147, 2001.

[39] J Vermeer. A geometric interpretation of ungar's addition and of gyration in the hyperbolic plane. *Topology and its Applications*, 152(3):226–242, 2005.

[40] Nisheeth K Vishnoi. Geodesic convex optimization: Differentiation on manifolds, geodesics, and convexity. *arXiv preprint arXiv:1806.06373*, 2018.

[41] Xuan Wei, Yun Wei, and Xiaobo Lu. RMDC:Rotation-mask deformable convolution for object detection in top-view fisheye cameras. *Neurocomputing*, 504:99–108, 2022.

[42] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Shuai Wu and Yong Xu. Dsn: A new deformable subnetwork for object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):2057–2066, 2019.

[44] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing*, 30:1169–1179, 2020.

[45] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. *Proceedings of the European conference on computer vision (ECCV)*, pages 469–484, 2018.

[46] Senthil Yogamani and et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9307–9317, 2019.

[47] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129:3051–3068, 2021.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[49] Keyao Zhao, Chunyu Lin, Kang Liao, Shangrong Yang, and Yao Zhao. Revisiting radial distortion rectification in polar-coordinates: A new and efficient learning perspective. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3552–3560, 2022.

[50] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019.

[51] Yucheng Zhu, Guangtao Zhai, Yiwei Yang, Huiyu Duan, Xiongkuo Min, and Xiaokang Yang. Viewing behavior supported visual saliency predictor for 360 degree videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4188–4201, 2021.