Sph2Pob: Boosting Object Detection on Spherical Images with Planar Oriented Boxes Methods

IJCAI/2023 MACAO

Github Codes

952 | Wed 23 August 11:45-12:45 (5 / 6)

Xinyuan Liu[1,2], Hang Xu[3], Bin Chen[1,2], Qiang Zhao[1], Yike Ma[1], Chenggang Yan[3], Feng Dai[1*]

[1]*Institute of Computing Technology, Chinese Academy of Sciences*

[2]*University of Chinese Academy of Sciences* [3]*Hangzhou Dianzi University*

❖ Introduction ❖

➤ **Task**

- detect (locate+classfy) objects on panoramic / spherical images.

➤ **Application**

- environment perception for robotics and automatic driving.

➤ **Focus point**

- IoU calculation and loss design for spherical bounding boxes.

➤ **Challenges**

- It's hard to balance differnetiablity, accuary and speed in spherical IoU calculation.
- It's hard to design better spherical loss functions beyond naive L1-Loss.

➤ **Contributions**

- convert spherical boxes into planar oriented boxes in pairs, named as Sph2Pob.
- implement a differentiable, fast, accurate spherical IoU based on Sph2Pob.
- implement a flexible and extensible spherical loss functions based on Sph2Pob.

❖ Prerequisites ❖

➤ **Spherical Images**

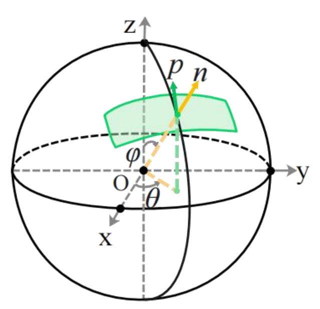
- Spherical image is a natural extend (360° view) of comon planar image.
- Spherical image has two display mode, i.e., sphere and ERP.

Sphere

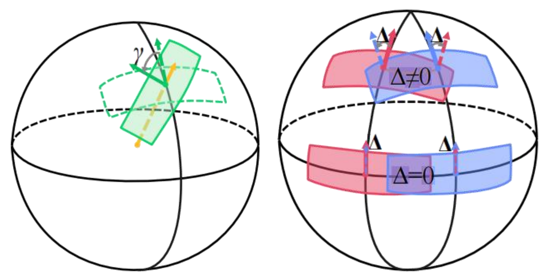
Equal Rectangular Projection (ERP)

➤ **Spherical Boxes**

- Spherical bounding box is defined as $(\theta, \phi, \alpha, \beta, \gamma)$.
- $n(\theta, \phi)$ is the tangent point of the sphere and rectangular tangent plane.
- α and β are the horizontal and vertical fields of view of the spherical bounding box.



- γ is rotated angle around center-axis $p(\theta, \phi)$.
- Apart from γ , another rotated angle Δ coupled with box-pair exists on sphere. **[our insight]**
- call γ as *external angle*, while Δ as *internal angle*.



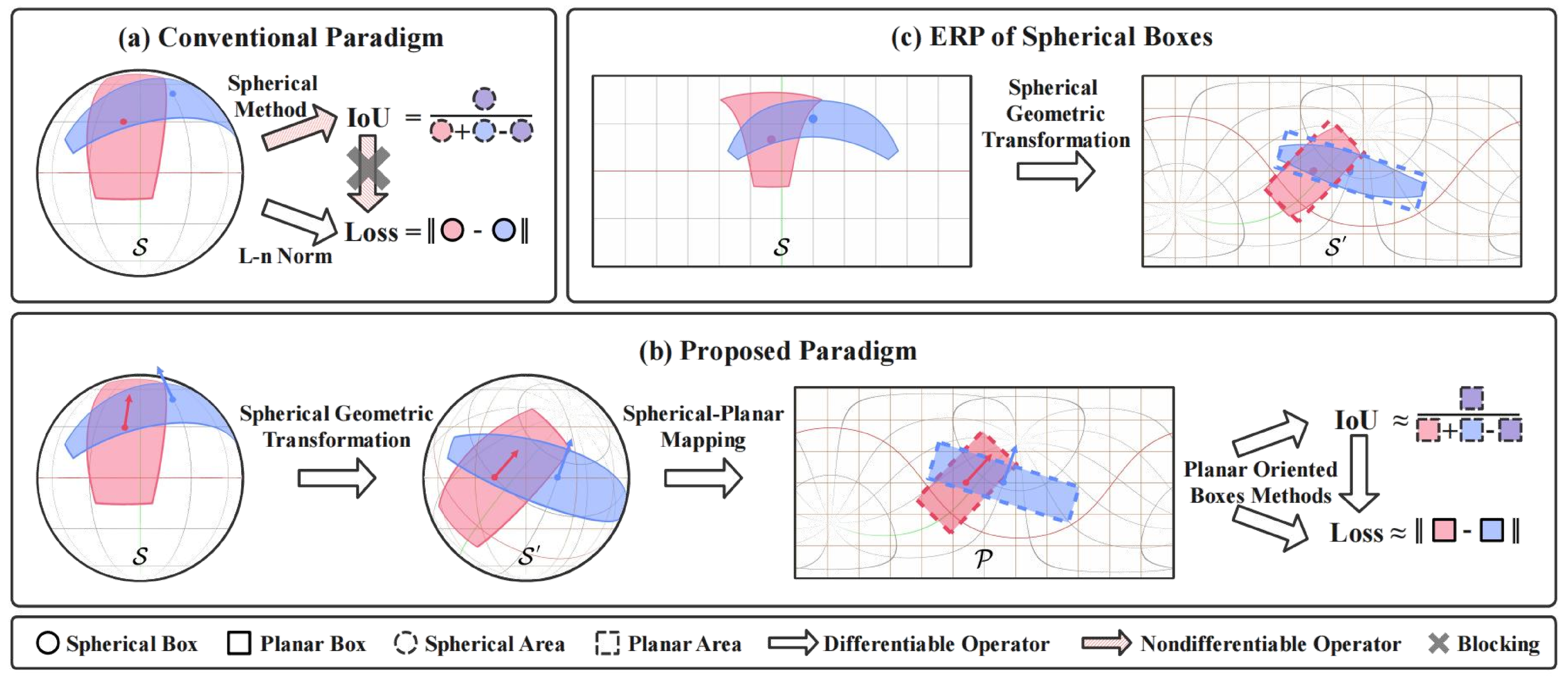
external angle *internal angle*

Welcome to communicate with me through Email !

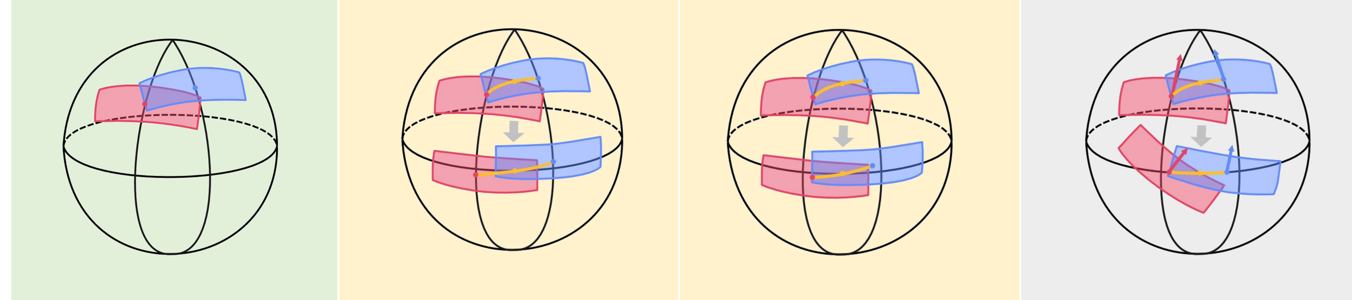
liuxinyuan21s@ict.ac.cn ***fdai@ict.ac.cn***

❖ Methodology ❖

➤ **Overview**



➤ **Comparsion**



	Exact Method	Approximate Method		Approximate Method
Method	Unbiased-IoU	Sph-Io2U	FoV-IoU	Sph2Pob-IoU(Our)
differentiablity	☆☆☆☆☆	★★★★★	★★★★★	★★★★★
speed	★☆☆☆☆	★★★★★	★★★★★	★★★★☆
accuracy	★★★★★	★☆☆☆☆	★★☆☆☆	★★★★☆

➤ **Mathematical Details**

1. Compute position and pose.

$$\boldsymbol{n} = \boldsymbol{n}(\theta, \phi) = [nx, ny, nz]^\top = [\sin(\phi)\cos(\theta), \sin(\phi)\sin(\theta), \cos(\phi)]^\top$$

$$\boldsymbol{p} = \boldsymbol{p}(\theta, \phi) = \frac{\partial \boldsymbol{n}(\theta, \phi)}{\partial \phi} = [px, py, pz]^\top = [\cos(\phi)\cos(\theta), \cos(\phi)\sin(\theta), -\sin(\phi)]^\top$$

2. Construct spherical transformation.

$$\boldsymbol{R} = [\boldsymbol{v}_x, \boldsymbol{v}_y, \boldsymbol{v}_z]^\top = \left[\frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{\|\boldsymbol{n}_1+\boldsymbol{n}_2\|}, \frac{\boldsymbol{n}_1-\boldsymbol{n}_2}{\|\boldsymbol{n}_1-\boldsymbol{n}_2\|}, \boldsymbol{v}_x \times \boldsymbol{v}_y\right]^\top$$

3. Transform position and pose.

$$\hat{\boldsymbol{n}}(\hat{\theta}, \hat{\phi}) = \boldsymbol{R}\,\boldsymbol{n}(\theta, \phi) \qquad [\hat{\boldsymbol{p}}_1, \hat{\boldsymbol{p}}_2] = \boldsymbol{R}\,[\boldsymbol{p}_1, \boldsymbol{p}_2]$$

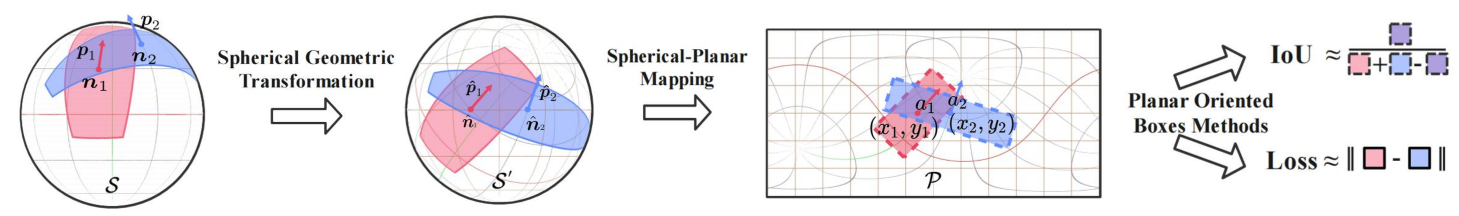
4. Compute *Internal Angle*.

$$\Delta = \Delta_1 + \Delta_2 = \arccos(\hat{\boldsymbol{p}}_1 \cdot \hat{\boldsymbol{p}}_{ref}) + \arccos(\hat{\boldsymbol{p}}_2 \cdot \hat{\boldsymbol{p}}_{ref})$$

5. Map spherical boxes to planar boxes.

$$\mathcal{B}_i^{\mathcal{P}} = (x_i, y_i, w_i, h_i, a_i) = (\hat{\theta}_i, \hat{\phi}_i, \hat{\alpha}_i, \hat{\beta}_i, \hat{\Delta}_i)$$

$$a_i = \Delta_i + \gamma_i, i = 1, 2$$

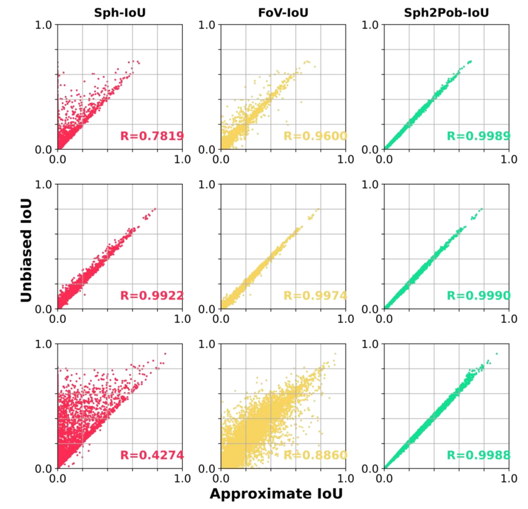


★ **IoU & Loss** ★ $IoU^{\mathcal{S}}(\boldsymbol{B}_1^{\mathcal{S}}, \boldsymbol{B}_2^{\mathcal{S}}) \approx IoU^{\mathcal{P}}(Sph2Pob(\boldsymbol{B}_1^{\mathcal{S}}, \boldsymbol{B}_2^{\mathcal{S}}))$ $Loss^{\mathcal{S}}(\boldsymbol{B}_1^{\mathcal{S}}, \boldsymbol{B}_2^{\mathcal{S}}) \approx Loss^{\mathcal{P}}(Sph2Pob(\boldsymbol{B}_1^{\mathcal{S}}, \boldsymbol{B}_2^{\mathcal{S}}))$

❖ Experiments ❖

➤ **Main Results**

- Scatter of different IoU.



- Comprehensive comparison of box transform methods.

Method	Consistency R_{all}↑	Consistency R_{low}↑	Consistency R_{high}↑	Time-cost T_{cpu}↓	Time-cost T_{cuda}↓	Detection AP↑	Detection AP_{50}↑	Detection AP_{75}↑
Sph	0.7819	0.9922	0.4274	0.0364	0.0033	10.7	24.3	7.8
Fov	0.9600	0.9974	0.8860	0.0372	0.0034	10.9	25.0	7.9
Sph2Pob	0.9989	0.9990	0.9988	2.2275	0.0096	11.5	25.7	8.2
Unbiased	1.0000	1.0000	1.0000	46.4417	-	-	-	-

- Ablation studies about edge & angle calculation.

Edge	Error↓(mean±std)	R↑
arc	0.0016±0.0042	0.9989
chord	0.0023±0.0063	0.9974
tangent	0.0086±0.0192	0.9681

Angle	Error↓(mean±std)	R↑
original	0.0025±0.0086	0.9946
equator	0.0016±0.0042	0.9989
project	0.0017±0.0043	0.9987

➤ **Evaluations**

- Evaluation on different Loss.

Loss	360-Indoor AP↑	360-Indoor AP_{50}↑	360-Indoor AP_{75}↑	PANDORA AP↑	PANDORA AP_{50}↑	PANDORA AP_{75}↑
L1	10.2	23.0	7.8	10.3	24.3	6.6
L1†	9.9	21.9	7.7	10.1	23.7	6.8
GWD† [Yang *et al.*, 2021b]	6.8	14.5	5.6	5.9	12.3	5.0
KLD† [Yang *et al.*, 2021c]	9.5	21.5	6.8	10.3	23.5	7.1
KFIoU† [Yang *et al.*, 2022b]	8.5	19.7	6.2	9.6	23.2	5.6
IoU† [Yu *et al.*, 2016]	9.8	22.1	6.8	10.4	24.8	6.9
GIoU† [Rezatofighi *et al.*, 2019]	10.5	23.9	7.8	10.3	24.7	6.8
DIoU† [Zheng *et al.*, 2020]	11.0	24.6	8.2	10.4	24.8	7.0
CIoU† [Zheng *et al.*, 2021]	11.5	25.7	8.2	10.5	25.3	7.0

- Evaluation on different detectors.

Detector	Loss	360-Indoor AP↑	360-Indoor AP_{50}↑	360-Indoor AP_{75}↑	PANDORA AP↑	PANDORA AP_{50}↑	PANDORA AP_{75}↑
Faster R-CNN	L1	12.5	28.1	9.1	11.0	27.8	6.2
	CIoU†	12.9	29.1	9.4	11.3	28.6	7.1
SSD	L1	10.8	27.6	6.3	9.5	25.8	4.6
	CIoU†	12.0	28.7	8.0	10.5	26.9	6.0
FCOS	L1	8.8	20.2	6.7	7.7	19.7	4.4
	CIoU†	9.2	21.0	7.0	8.8	21.2	5.6

❖ Visualization ❖