Unbiased IoU for Spherical Image Object Detection

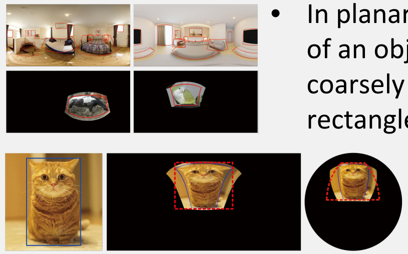
Feng Dai[1], Bin Chen[1,2], Hang Xu[3], Yike Ma[1], Xiaodong Li[4], Bailan Feng[4], Peng Yuan[4], Chenggang Yan[3], Qiang Zhao[1]*

[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

[2]University of Chinese Academy of Sciences, Beijing, China [3]Hangzhou Dianzi University, Hangzhou, China [4]Noah's Ark Lab, Huawei, China

AAAI-22

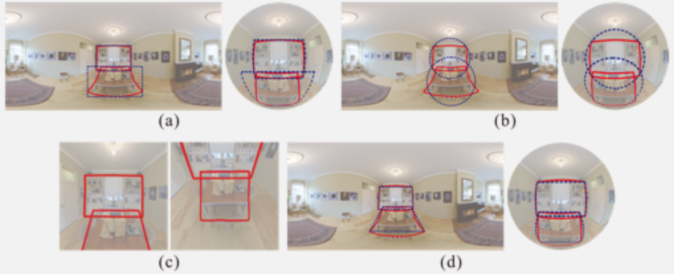
Introduction


(a) (b) (c)

- In planar images, the location of an object are defined coarsely using an axis-aligned rectangle (x, y, w, h) , and it is not suitable and can not tightly bound the distorted objects in spherical images.
- The existing methods give the excessive approximate IoU calculations, and the incorrect results will lead to poor performance and the unplausible evaluation for spherical image object detection task.

Existing Biased Evaluation Criteria

Red Curve: Sph. Rectangles; Blue Curve: Biased Criteria


(a) (b) (c) (d)

- **Biased Bounding Boxes as Representations:**

(a) Using axis-aligned rectangles on spherical images;

(b) Using circles on spherical images;

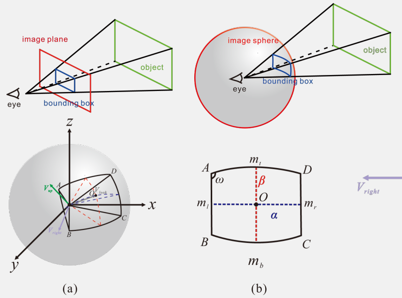
- **Biased Approximate Calculations:**

(c) Using axis-aligned rectangles on tangent planes;

(d) Using sampled spaced points on tangent planes but computing IoUs based on polygons on spherical images.

Note! All existing criteria are unreasonable because of either biased representations or biased calculations.

Unbiased IoU



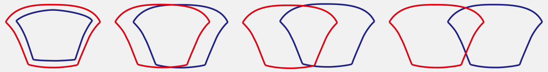
1. Area of Spherical Rectangle

We use (θ, ϕ, α, β) as unbiased representations for spherical rectangles, which corresponds to the azimuthal and polar angles and the FOVs.

After derivation, the area of each spherical rectangle is given by

$$A(b) = 4\arccos\left(-\sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\right) - 2\pi$$

2. Intersection Area Computation Algorithm


Different Shapes of intersection areas

- **Step 7&8.** Compute normal vectors and point vectors for eight boundaries by cross product;
- **Step 10&11.** Remove points outside the two spherical rectangles by dot product and redundant points by loop detection, which uses DFS Algorithm to find a closed loop.
- **Step 13&15.** Compute all left angles and the intersection area by $A(b_1 \cap b_2) = \sum_{i=1}^{n} \omega_i - (n-2)\pi$.

Algorithm 1: Intersection Area Computation

Input: Two spherical rectangles b_1 and b_2 denoted as $(\theta_1, \phi_1, \alpha_1, \beta_1)$ and $(\theta_2, \phi_2, \alpha_2, \beta_2)$

Output: the area of intersection $A(b_1 \cap b_2)$

1 **if** $b_1 \cap b_2 = \emptyset$ **then**

2 | **return** 0;

3 **end**

4 **if** $b_1 \subset b_2$ or $b_2 \subset b_1$ **then**

5 | **return** $\min(A(b_1), A(b_2))$;

6 **end**

7 compute the vertices $\mathcal{V}_i$ of spherical rectangle b_i;

8 compute the set $\mathcal{P}$ of intersection points between boundaries of b_1 and those of b_2;

9 $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{V}_1 \cup \mathcal{V}_2$;

10 remove the points p in $\mathcal{P}$ such that $p \notin b_1$ or $p \notin b_2$;

11 remove duplicated points in $\mathcal{P}$ via loop detection;

12 **for** $p_i \in \mathcal{P}$ **do**

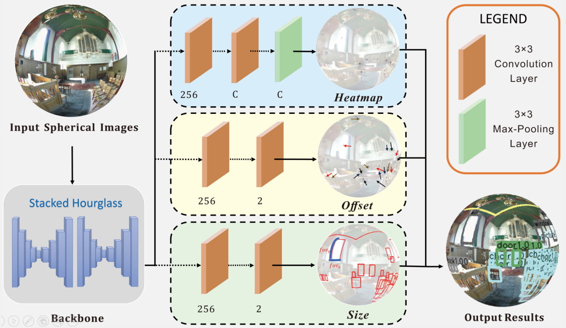
13 | compute the angle ω_i

14 **end**

15 **return** $A(b_1 \cap b_2)$ computed via Equation 3;

Intersection Computation Algorithm

Spherical CenterNet



- **Classification Loss**

$$L_{cls} = -\frac{1}{N}\sum_{xyc} w_{xy} \begin{cases} (1-p_{xyc})^2 \log(p_{xyc}) & \text{if } y_{xyc}=1, \\ (1-y_{xyc})^4 (p_{xyc})^2 \log(1-p_{xyc}) & \text{otherwise.} \end{cases}$$

$$w_{xy} = \left(\cos\frac{y\pi}{H} - \cos\frac{(y+1)\pi}{H}\right)\frac{2\pi}{W}$$

- **Offset Regression Loss**

$$L_{off} = \frac{1}{N}\sum_i \arccos\left(\langle \Gamma(c_i + o_i), \Gamma(c_i + \hat{o}_i) \rangle\right)$$

- **FOVs Regression Loss**

$$L_{fov} = \frac{1}{N}\sum_i |s_i - \hat{s}_i|$$

- **Total Loss**

$$L = L_{cls} + \lambda_{off} L_{off} + \lambda_{fov} L_{fov}$$

Experiments & Results

Cases	Methods	IoUs	△
	Sph. Integral	0.32006	-
	Rectangle	0.47163	0.15157
	Polygon	0.35891	0.03885
	Circle	0.24286	0.07720
	SphIoU	0.16537	0.15469
	Ours	**0.31974**	**0.00032**
	Sph. Integral	0.25801	-
	Rectangle	0.55155	0.29354
	Polygon	0.26958	0.01157
	Circle	0.24996	0.00805
	SphIoU	0.11392	0.17109
	Ours	**0.25772**	**0.00029**
	Sph. Integral	0.33966	-
	Rectangle	0.25870	0.08096
	Polygon	0.31526	0.02440
	Circle	0.35992	0.02026
	SphIoU	0.34220	0.00254
	Ours	**0.33935**	**0.00031**

- Our unbiased IoU compared to other biased representations
- Our unbiased IoU compared to Spherical Integral Method

Cases	Methods	$12k \times 6k$	$10k \times 5k$	$8k \times 4k$
	Sph. Integral	0.32006	0.32012	0.32022
	Ours	0.31974	0.31974	0.31974
	△	**0.00032**	**0.00038**	**0.00048**
	Sph. Integral	0.25801	0.25807	0.25816
	Ours	0.25772	0.25772	0.25772
	△	**0.00029**	**0.00035**	**0.00044**
	Sph. Integral	0.33966	0.33972	0.33981
	Ours	0.33935	0.33935	0.33935
	△	**0.00031**	**0.00037**	**0.00046**

Detection Results compered to the other methods

Methods	Backbone	360-Indoor			360-VOC-Gaussian			360-VOC-Uniform		
		AP	AP^{50}	AP^{75}	AP	AP^{50}	AP^{75}	AP	AP^{50}	AP^{75}
CenterNet	ResNet-101	8.6	20.5	5.8	43.3	81.9	40.3	8.3	14.1	8.8
Multi-Kernel	ResNet-101	4.7	11.1	2.8	55.9	77.7	64.8	7.0	12.5	7.3
Sphere-SSD	ResNet-101	2.9	7.8	1.4	21.8	28.4	26.7	11.7	19.2	13.4
Reprojection R-CNN	ResNet-101	5.0	15.3	1.9	53.6	62.2	44.8	9.5	13.8	10.1
Ours	ResNet-101	**10.0**	**24.8**	**6.0**	**65.5**	**84.6**	**75.5**	**15.8**	**21.5**	**18.1**

Visualization

Ground Truth Heatmap

Spherical Object Detection Results Visualization