Spherical Criteria for Fast and Accurate 360° Object Detection

Pengyu Zhao,* Ansheng You,* Yuanxing Zhang,* Jiaying Liu, Kaigui Bian, Yunhai Tong
School of EECS, Peking University, Beijing, China
{pengyuzhao, youansheng, longo, liujiaying, bkg, yhtong}@pku.edu.cn

Abstract

With the advance of omnidirectional panoramic technology, 360° imagery has become increasingly popular in the past few years. To better understand the 360° content, many works resort to the 360° object detection and various criteria have been proposed to bound the objects and compute the intersection-over-union (IoU) between bounding boxes based on the common equirectangular projection (ERP) or perspective projection (PSP). However, the existing 360° criteria are either inaccurate or inefficient for real-world scenarios. In this paper, we introduce a novel *spherical criteria* for fast and accurate 360° object detection, including both spherical bounding boxes and spherical IoU (SphIoU). Based on the spherical criteria, we propose a novel two-stage 360° detector, i.e., *Reprojection R-CNN*, by combining the advantages of both ERP and PSP, yielding efficient and accurate 360° object detection. To validate the design of spherical criteria and Reprojection R-CNN, we construct two unbiased synthetic datasets for training and evaluation. Experimental results reveal that compared with the existing criteria, the two-stage detector with spherical criteria achieves the best mAP results under the same inference speed, demonstrating that the spherical criteria can be more suitable for 360° object detection. Moreover, Reprojection R-CNN outperforms the previous state-of-the-art methods by over 30% on mAP with competitive speed, which confirms the efficiency and accuracy of the design.

1 Introduction

In the past few years, virtual reality techniques have developed rapidly owing to the development of 360° cameras with omnidirectional vision. The 360° images and videos allow users to receive detailed information, thereby improving the quality of experiences (Ardouin et al. 2012; Huang et al. 2017). 360° cameras also play important roles in scenarios which require wide-range field-of-view (FoV). *Object detection* is a significant computer vision task that deals with detecting semantic objects in images and videos. Recent advances with convolutional neural network (CNN) (Ren et al. 2015; Liu et al. 2016; He et al. 2017) achieve remarkable improvements in 2D task. However, object detection in *360° images* is still challenging due to the following two reasons, as listed below.

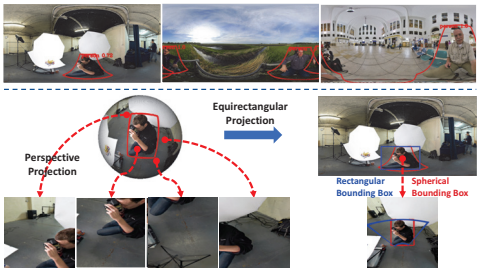

Figure 1: Challenges in 360° object detection. Objects in ERP suffer from distortion and discontinuity on the borders while objects can hardly be recognized with only few PSPs. Besides, spherical BB (red outline) bounds the object more tightly than common rectangular BB (blue outline)

Lack of Appropriate Criteria. Unlike image classification task, additional information is required in object detection to locate objects and compute metrics, i.e., *bounding boxes* (BB) and *intersection-over-union* (IoU). Although existing works proposed various criteria for 360° object detection, they either introduce bias in BB and IoU (Yang et al. 2018; Lee et al. 2019; Yu and Ji 2019; Wang and Lai 2019), or could not efficiently compute IoU (Coors, Paul Condurache, and Geiger 2018). Moreover, some criteria (Su and Grauman 2017) could not even be applied in the actual scenarios. Thus, it is necessary to introduce an efficient and accurate criteria for 360° object detection.

Dilemma between Distortion Reduction and Efficiency. 360° images are typically represented by *equirectangular projection* (ERP) (Snyder 1997) or multiple *perspective projections* (PSP). ERP is generated by polar transformation, and thus suffers from both distortion in the polar regions and discontinuity on the boundary. PSP projects a partial area of the sphere onto a focal plane with little distortion but a large number of candidate areas are required to cover all the objects on sphere, which is time consuming.

To address the above challenges, we explore the unbiased criteria on sphere and introduce *spherical criteria* as

*The first three authors contribute equally to this paper.

an **efficient** and **accurate** approximation of unbiased measurement, including both *spherical BB* (SphBB) and *spherical IoU* (SphIoU). Based on spherical criteria, we propose a novel **two-stage** object detector, *i.e.*, *Reprojection R-CNN* (Rep R-CNN), by taking full advantage of **both** ERP and PSP. Specifically, it generates candidate regions efficiently based on the omnidirectional FoV of ERP and conducts precise refinement over the distortion-free PSPs. Due to the lack of **unbiased** dataset for 360° object detection, we construct two datasets, i.e., *VOC360* and *COCO-Men*, for training and evaluation. Experiment results reveal that spherical criteria can lead to more accurate predictions than the existing criteria on the same baseline model with the similar inference speed, demonstrating both efficiency and accuracy of the design. Moreover, Rep R-CNN outperforms all state-of-the-art methods on both datasets and achieves at least 30% improvement over the strongest baseline. Besides, Rep R-CNN gains good detection results in realistic scenarios, indicating the feasibility for some real-world applications.

2 Related Work

CNNs on 360° Vision: Recent advances in 360° images resort to geometric information on the sphere. (Khasanova and Frossard 2017) applies graph convolutional network on ERP. (Esteves et al. 2018) proposes SO(3) 3D rotation group in convolutions and on top of that, suggests transforming S2 space to a SO(3) representation to reduce distortion and encode rotation equivariance in the network. also introduces spherical U-Net for saliency detection based on spherical property. Meanwhile, some works attempt to directly solve the distortion in ERP. (Su and Grauman 2017; 2019) transfer knowledge from a pre-trained CNN on PSP to novel networks on ERP. Other approaches (Coors, Paul Condurache, and Geiger 2018; Tateno, Navab, and Tombari 2018; Zhao et al. 2018) refer to the idea of deformable convolution (Dai et al. 2017), and propose the distortion-aware spherical convolution (SphConv), where the convolutional filter gets distorted in the same way as the objects on ERP. Though the above methods outperform planar CNNs, they could not completely eliminate the distortion on ERP. The representations other than ERP and PSP, e.g., Cubemap (Boomsma and Frellsen 2017; Cheng et al. 2018) and icosahedral mesh (Lee et al. 2019; Cohen et al. 2019; Jiang et al. 2019) are also introduced for 360° images. Since they still suffer from distortion and heavy computation latency, they are not widely used in the real-world scenarios.

Object Detection in 360° Images: (Su and Grauman 2017) introduces SPHCNN as the backbone network of Faster R-CNN (Ren et al. 2015). For evaluation, they construct a synthetic dataset by projecting BBs in 2D images onto the sphere, and assume the object centers are known in advance. (Yang et al. 2018) exploits a PSP-based YOLO detector (Redmon and Farhadi 2017) on a real-world 360° dataset. They frame the objects with rectangular BBs on ERP, which are distorted on sphere. (Yu and Ji 2019; Wang and Lai 2019) also follow this simple ERP-oriented criteria so that the predictions are still biased even if novel convolutional kernels are applied. In contrast, (Coors, Paul Condurache, and Geiger 2018) utilizes a tangent plane based BB for unbiased 360° detection. However, since it is hard to measure BBs on different tangent planes, the IoU computation is still defined on distorted ERP. They also build up the synthetic FlyingCars dataset for experiment. Different from above methods, SpherePHD (Lee et al. 2019) conducts detection on polyhedrons in the SYNTHIA dataset (Ros et al. 2016), and uses bounding circles as measurements.

Figure 2: ERPBB (blue outline), CirBB (orange outline) and UnbBB (TanBB/SphBB, red outline) on sphere and ERP.

It is apparent that the existing methods exploit various biased criteria, i.e., BBs and IoUs, in different datasets. This situation attributes to the lack of appropriate 360° criteria and 360° dataset. In this paper, we introduce efficient and accurate spherical criteria for practical 360° object detection, and create two datasets for unbiased training and evaluation.

3 Spherical Criteria for Fast and Accurate 360° Object Detection

Criteria of 360° Object Detection

The BB and IoU are fundamental part of object detection, where the mean average precision (mAP) and the widely-used non-maximum-suppression (NMS) are all defined on those two elements. In the scenario of 360° object detection, a normal rectangular BB can not appropriately bound the object on sphere, and it is also difficult to define the intersection between objects with different centers due to the curvature of sphere. Therefore, it is necessary to establish a standard (unbiased) criteria for 360° object detection.

Unbiased Criterion for 360° Object Detection

Suppose you are watching a VR video, where each frame is a 360° image, and you want to see an object clearly on the image. You turn the body and move the head until your sight is aligned with the object center. Then, the object is in the center of your field-of-view and forms a curved rectangle on the sphere. Following the above procedure, we can deduce that the unbiased BB on sphere (UnbBB) can be represented by either a certain part of FoV on sphere or a rectangle on the tangent plane where the object center is the tangent point. Note that these two representations are equivalent as shown in Figure 2, and are both unbiased. However, it is still hard to directly define the intersection between UnbBBs because: (1) The center of the intersection can not be properly defined when UnbBBs have different centers; (2) The shape of intersection is usually irregular on sphere. To ensure an unbiased and uniform measurement, we compute unbiased IoU (UnbIoU) by integral on sphere. The UnbBB and UnbIoU are served as the ground-truth criteria in this paper.

Existing Criteria for 360° Object Detection

Before going into the proposed spherical criteria, we will first look into the existing criteria on 360° object detection:

1. The existing methods (Yang et al. 2018; Yu and Ji 2019; Wang and Lai 2019) mainly regard ERP as a 2D image, and bound objects by rectangles on ERP. Since ERP is unwrapped by polar coordinates, the BB of object B on ERP (ERPBB) can be represented by: $ERPBB(B) = (B_\theta, B_\phi, B_{\Delta\theta}, B_{\Delta\phi})$, where B_θ and B_ϕ represent the latitude/longitude of the object's center, and $B_{\Delta\theta}$, $B_{\Delta\phi}$ represent the transformed width/height of the object's occupation on ERP. Similarly, the IoU on ERP (ERPIoU) is computed in the same way as the planar detection task.
2. (Su and Grauman 2017) and (Coors, Paul Condurache, and Geiger 2018) utilize the rectangle on the tangent plane as BB (TanBB) annotation: $TanBB(B) = (B_\theta, B_\phi, B_w, B_h)$, where B_θ and B_ϕ are the object's center, while B_w and B_h are the width/height of rectangle. Besides, Coors et al. also introduce in-plane rotation in TanBB but it does not match the realistic scenario. Since only considers the IoU between proposals with the same center point (since they do not aim at building a detector), we refer to the method proposed by Coors et al., where IoU (PolyIoU) is approximated by the overlap of two polygonal regions on ERP. Specifically, the outlines of TanBBs are evenly sampled on tangent planes and projected by inverse gnomonic projection (Coxeter 1961), forming convex polygons on ERP. The PolyIoU can then be linearly computed by (Preparata and Shamos 2012).
3. (Lee et al. 2019) exploits circular BB (CirBB) on ERP, cubemap and SphPHD to bound the objects. Here, we only consider CirBB on ERP because it is best-performed in the natural non-rotation task (Lee et al. 2019) and is more common than cubemap and SphPHD. A CirBB can be represented by: $CirBB(B) = (B_\theta, B_\phi, r)$, where r is the radius. The IoU between CirBBs (CirIoU) is computed based on the circle-circle intersection (Weisstein 2003).

Limitation of Unbiased and Existing Criteria

In this section we will show the advantages and drawbacks of the existing criteria for the practical 360° object detection.

1. UnbBB and UnbIoU: It is undoubted that unbiased criteria would lead to the most accurate 360° object detection. However, the computation of UnbIoU is time-consuming that is inefficient for practical 360° detection task.
2. ERPBB and ERPIoU: Due to the uneven sampling of polar projection, the pixel size on ERP varies with latitude and thus the uniform ERP-based measurement is biased on sphere, as illustrated in Figure 2. Besides, an identical ERPBB may correspond to areas of various shapes and sizes at different locations on the sphere.
3. TanBB and PolyIoU: TanBB is shown to be unbiased such that it could tightly bound the objects on the sphere. However, although PolyIoU is a more accurate measurement of UnbIoU, it still suffers from the bias on ERP and requires more time in the projection and IoU computing.

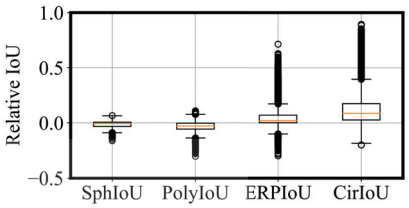

Figure 3: Relative IoU between UnbIoU and other IoUs.

4. CirBB and CirIoU: CirBB and CirIoU sustain the more series bias problem than ERP-based criteria as shown in Figure 2. Besides, CirBB may exceed the upper/lower boundaries of ERP when the objects are near the pole.

Spherical Criteria for Fast and Accurate 360° Object Detection

According to the above analysis, we can conclude that the existing criteria are either biased or inefficient for real-world detection task. Thus, we introduce the *spherical criteria* for fast and accurate 360° object detection, including both *spherical BB* (SphBB) and *spherical IoU* (SphIoU).

Spherical BB: To eliminate the bias in ERP, we utilize the unbiased FoVs to measure the size of objects: $SphBB(B) = (B_\theta, B_\phi, B_{fov_x}, B_{fov_y})$, where B_θ and B_ϕ denote the object center, and B_{fov_x}, B_{fov_y} represent the left-right/up-down FoVs of the object's occupation. Note that SphBB is directly defined on sphere other than ERP or tangent plane. The shape of SphBB can be regarded as a portion of spherical segment (Kern and Bland 1938) centered at the equator. In concrete, assume the SphBB of B is moved to the equator, which does not influence the shape and size, and then the area on the unit ball can be computed as:

$$\begin{aligned}\text{Area}(B) &= 2\pi r \cdot h \cdot p && (1)\\ &= 2\pi r \cdot 2r\sin(B_{fov_y}/2) \cdot (B_{fov_x}/2\pi) && (2)\\ &= 2r^2 B_{fov_x}\sin(B_{fov_y}/2) && (3)\\ &= 2B_{fov_x}\sin(B_{fov_y}/2). && (4)\end{aligned}$$

Spherical IoU: Based on SphBB, we introduce SphIoU for fast and accurate approximation of UnbIoU. SphIoU assumes that the intersection between two SphBBs B^i and B^j also forms a SphBB. The FoVs of the intersection can then be derived from the difference between the upper left and lower right corners of the rectangle, which is similar to the planar IoU, except that the width and height are now determined by FoVs. Then, the FoVs of the intersection, namely $B^{ij}_{fov_x}$ and $B^{ij}_{fov_y}$, can be deduced by:

$$\delta^{ij}_{x_{\max}} = \min\{B^i_\phi + B^i_{fov_x}/2, B^j_\phi + B^j_{fov_x}/2\}, \quad (5)$$

$$\delta^{ij}_{x_{\min}} = \max\{B^i_\phi - B^i_{fov_x}/2, B^j_\phi - B^j_{fov_x}/2\}, \quad (6)$$

$$\delta^{ij}_{y_{\max}} = \min\{B^i_\theta + B^i_{fov_y}/2, B^j_\theta + B^j_{fov_y}/2\}, \quad (7)$$

$$\delta^{ij}_{y_{\min}} = \max\{B^i_\theta - B^i_{fov_y}/2, B^j_\theta - B^j_{fov_y}/2\}, \quad (8)$$

$$B^{ij}_{fov_x} = \max\{0, \delta^{ij}_{x_{\max}} - \delta^{ij}_{x_{\min}}\}, \quad (9)$$

$$B^{ij}_{fov_y} = \max\{0, \delta^{ij}_{y_{\max}} - \delta^{ij}_{y_{\min}}\}. \quad (10)$$

It is worth mentioning that the above computation is equivalent to the case that we first move both B_i and B_j to the equator (by adding a constant on the latitudes because any change on the latitudes do not affect the final result in Eqn. 9 and Eqn. 10), and then compute the approximated area of the intersection. Thus, SphIoU can exclude the case that a SphBB wraps the pole where the intersection is hard to define, and also possess a more accurate estimation of actual IoU because the pixel size on equator is more uniform. Besides, the centers of B^i and B^j may appear in separate boundaries of ERP where SphBBs are far in the polar coordinates (e.g., -175° and 175° in longitude), resulting in zero IoU, but cover similar regions on the sphere. Thus, we rotate the sphere by 180° along z-axis and compute a rotated IoU between B^i and B^j. It can be inferred that the latitudes remain unchanged, and the longitudes become $(B^{i'}_\phi = B^i_\phi \,\%\, 360 - 180)$ and $(B^{j'}_\phi = B^j_\phi \,\%\, 360 - 180)$. Then, we take the maximum of the origin IoU and the rotated IoU as the final SphIoU.

Advantages of Proposed Spherical Criteria

In this section, we will show the advantages of the proposed spherical criteria by comparing it with other criteria. Specifically, we exploit IoU accuracy and IoU computation complexity as the measurement index.

IoU Accuracy: To measure the IoU accuracy, we randomly select BBs on sphere, compute IoUs in different criteria between pairs of BBs. Then, We calculate the relative IoU difference between UnbIoU and the other IoUs, and plot the diagram in Figure 3. Since both ERPBB and CirBB use biased BBs on ERP and introduce extreme bias in the IoU measurement, ERPIoU and CirIoU are much more inaccurate than SphIoU and PolyIoU. Though PolyIoU utilize unbiased TanBB to frame the objects, it still suffers from distortion on ERP. Owing to the spherical measurement of SphBB and the exclusion of extreme situations, the proposed SphIoU has less than 0.1 deviation from the UnbIoU, indicating that SphIoU is an accurate approximation of ground truth.

IoU Computation Complexity: It is obvious that ERPIoU, CirIoU and SphIoU can all be computed in O(1) time. Though PolyIoU needs to re-project the sampled outlines back to ERP, the projections can be pre-calculated and thus only linear computation on the number of outline sampling is required for PolyIoU. Different from the above IoUs, the UnbIoU is highly time-consuming due to the integral on sphere, which is inefficient for real-world 360° applications.

4 Reprojection R-CNN

Two-stage 360° Detection with Spherical Criteria

It is a common consensus that two-stage detectors can reach higher accuracy rates, but are inherent slower than one-stage detectors. Thus, for a practical two-stage 360° detector, it is necessary to deal with the speed problem while maintaining the accuracy. To compensate for the low speed, the existing two-stage 360° detectors (Yu and Ji 2019; Wang and Lai 2019) adopt the less computationally intensive ERP-based criteria to frame the objects, but introduce severe bias and distortion in the detection results. Regarding the proposed spherical criteria, it does not introduce any bias in BB and attains relatively high speed in IoU computation under little precision sacrifice, which is necessary for fast and accurate two-stage 360° object detection. Thus, we apply spherical criteria in the proposed two-stage detector.

Combining ERP and PSP in a Two-stage Detector

As discussed in Sec. 1, ERP and PSP are two common representations of 360° images and they have their own advantages and disadvantages. Though ERP introduces severe distortion in the image, it possesses 360° FoV which possesses all information on the sphere. Meanwhile, PSP could eliminate distortion with a large number of projections, but it is time-consuming to handle these projections.

The previous methods (Cohen et al. 2018; Yang et al. 2018; Yu and Ji 2019; Lee et al. 2019) only utilize one of the representations that either suffers from distortion of ERP or requires plenty of time with PSP. In contrast, we take the advantages of both representations by applying ERP and PSP as the inputs of a single two-stage 360° detector named *Reprojection R-CNN* (Rep R-CNN).

Architecture of Reprojection R-CNN

The overall architecture of Rep R-CNN is illustrated in Figure 4. Rep R-CNN contains two stages, where the first stage is a spherical RPN (SphRPN) that efficiently proposes coarse detections on ERP, and the second stage is a reprojection network (RepNet) that accurately refines the proposals based on PSPs. A reprojection RoI alignment (Rep RoIAlign) is introduced to bridge SphRPN and RepNet by transforming the SphBBs to the fixed-size inputs for RepNet. The precise architecture of Rep R-CNN is given below.

Spherical Region Proposal Network: Given the ERP of a 360° image, SphRPN generates the objectness score and the offset of SphBB for each candidate region. Different from vanilla RPN (Ren et al. 2015), SphRPN adopts SphConv (Coors, Paul Condurache, and Geiger 2018) in the backbone network to efficiently extract a distortion-aware feature map. SphConv adjusts the sampling locations of the convolutional filters by projecting uniform convolutional filters on the tangent planes centered at corresponding locations back to ERP via inverse transformation of gnomonic projection (Coxeter 1961; Snyder 1987). Since it only alters the sampling locations, it could extract a more accurate feature map at the same computation cost. Additionally, we also introduce *spherical anchors* (SphAnchor) as regression references. SphAnchor simply replaces the height/width measurement by FoVs (*e.g.*, $30° \times 60°$). SphNet predicts k corresponding BBs at each location on the feature map based on SphAnchors with various shapes and sizes.

Reprojection RoI Alignment: Given the BBs generated by SphRPN, Rep RoIAlign expands the predictions, reprojects the expanded areas on **raw pixels of ERP** to the tangent planes, and resizes the projections to fixed-size patches. Specifically, for each SphBB $(B_\theta, B_\phi, B_{fov_x}, B_{fov_y})$, since objects may be partially contained due to the biased sampling locations, Rep RoIAlign expands the FoVs of the

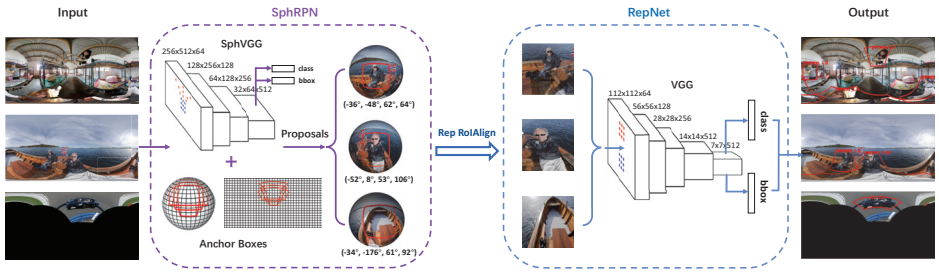

Figure 4: Architecture of Rep R-CNN: SphRPN employs SphRPN to generate coarse proposals; RepNet in the second stage applies a standard VGG backbone and yields precise SphBBs. A Rep RoIAlign layer is applied to bridge SphRPN and RepNet.

SphBB by a factor $r > 1$, yielding a larger SphBB as $(B_\theta, B_\phi, rB_{fov_x}, rB_{fov_y})$. Then, the expanded SphBB is reprojected to the tangent plane located at the predicted object center. For each point (θ, ϕ) within SphBB, the corresponding coordinates in tangent plane are calculated by gnomonic projection (Coxeter 1961; Snyder 1987):

$$f_x(\theta,\phi) = \frac{\cos\theta \, \sin(\phi - B_\phi)}{\sin B_\theta \, \sin\theta + \cos B_\theta \, \cos\theta \, \cos(\phi - B_\phi)}, \\ f_y(\theta,\phi) = \frac{\cos B_\theta \, \sin\theta - \sin B_\theta \, \cos\theta \, \cos(\phi - B_\phi)}{\sin B_\theta \, \sin\theta + \cos B_\theta \, \cos\theta \, \cos(\phi - B_\phi)}. \quad (11)$$

Then, Rep RoIAlign exploits the average RoI alignment and converts the obtained PSPs with various shapes and sizes into the patches with a fixed spatial extent.

Reprojection Network: Based on the distortion-free patches, RepNet simply applies another backbone network, i.e., a planar CNN on 2D images, to rectify each prediction, generating the final detection results for the 360° image.

Optimization

Loss Functions: We minimize a similar multi-task loss in both SphRPN and RepNet as Faster R-CNN (Ren et al. 2015). Both networks have two sibling output layers (Girshick 2015). Suppose that the concerned objects in the 360° images belong to a number of K categories. The first layer outputs the probability distribution over $K+1$ categories (including background), and the second layer outputs a *spherical bounding-box regression* offsets (use SphBBs as targets) parameterized by (Girshick et al. 2014) for each object class, i.e., $t = (t_\theta, t_\phi, t_{fov_x}, t_{fov_y})$. The loss function for regression t and the objectness score p is defined as:

$$L(p,t) = L_{cls}(p,p^*) + \lambda\,[p^* \geq 1] L_{reg}(t,t^*). \quad (12)$$

p^* is the ground-truth label, and t^* is the associated regression target. The classification loss L_{cls} is the log loss for true class, while the regression loss L_{reg} is the smooth L_1 loss in (Girshick 2015).

SphRPN: We set $K = 1$ in SphRPN, indicating whether the proposed regions belong to the foreground or background. Here, the references for the bounding-box regression are SphAnchors, which are assigned to foreground objects using a SphIoU threshold of 0.7, and to the background if the SphIoU is less than 0.3. We sample 128 positive and negative anchors per image with a ratio of 1:1. The balance parameter λ is set to 3 in all the experiments.

RepNet: Meanwhile, K is task-dependent in RepNet, and the references are SphBBs generated by SphRPN. The SphBB is now considered positive if the SphIoU between the prediction and the ground-truth achieves at least 0.5, and negative if SphIoU is less than 0.3. Besides, we sample 128 RoIs per image with a ratio of 1:3 of positive to negative, and set $\lambda = 1$ in RepNet.

Implementation Details

Backbone: We apply VGG-16 (Simonyan and Zisserman 2015) as the backbone network for both stages. conv5_3 is served as the final feature map of SphRPN.

Anchors: We use $k = 9$ anchors in SphRPN, with three scales of $(30°)^2$, $(60°)^2$ and $(90°)^2$ and three aspect ratios of 1:1, 1:2 and 2:1

Training: We train Rep R-CNN in two steps. In the first step, we initialize SphRPN by the model pre-trained on ImageNet dataset (Russakovsky et al. 2015), and fine-tune the network on specific 360° datasets with ERPs of size 512×1024 as input. In the second step, we adopt the regions proposed by SphRPN. The proposals are filtered by NMS with 0.7 threshold, and reprojected to fixed-size PSPs of 224×224 by Rep RoIAlign. We use the weights of SphRPN to initialize RepNet, but we do not share weights in backbone networks as we find that it would degrade performance. Both SphRPN and RepNet are trained on 4 GPUs for 20 epochs by 0.9 momentum optimizer, where the learning rate is initially set to 0.001 and then decreased by a factor of 10 after training 15 epochs. The batch size is 16 in SphRPN and 128 in RepNet.

Inference: At test time, we apply NMS with a threshold of 0.7 to reduce redundancy in SphRPN and select top-n ranked proposals for the second stage. After RepNet, we drop the proposals with less than 0.1 confidence score and apply another NMS of 0.45 to generate the final detections.

5 Experiments

Experimental Setup

Datasets: We conduct experiment on three datasets, including two novel synthetic datasets annotated by **UnbBB** (including both SphBB and TanBB) and one real-world dataset without pre-labeled annotation.

VOC360: VOC360 is a synthetic dataset generated from PASCAL VOC 2007 and 2012 (Everingham et al. 2010) with 20 categories. We crop the objects with random-sized

	n	VOC360	COCO-Men	Speed
UnbBB and UnbIoU	50	72.47	83.89	597ms
ERPBB and ERPIoU	50	30.61	37.79	163ms
CirBB and CirIoU	50	24.48	29.36	182ms
TanBB and PolyIoU	50	58.35	65.17	264ms
SphBB and SphIoU	50	71.65	81.48	178ms

Table 1: mAP and inference speed of different criteria in Rep R-CNN on both VOC360 and COCO-Men datasets. n represents the number of PSPs used in the network.

background from images, and then project the cropped images to arbitrary points on the sphere. Each image in VOC360 is attached by only **one** cropped image. VOC360 has 15000 training images, 1800 validation images, and 4955 test images.

COCO-Men: For the multi-object scenario, we construct COCO-Men dataset, which combines the real-world background 360° images and the segmented images of people cropped from COCO dataset (Lin et al. 2014). Each image includes **three to six** people, and every pair of people has an overlapping of less than 0.3. In total, the dataset comprises 4000 training images, 2000 validation images, and 1000 test images.

SUN360: To demonstrate the capability of Rep R-CNN for real scenes, we use SUN360 dataset (Xiao et al. 2012) which contains real-world 360° images from Internet.

Performance Metric: To give a comprehensive evaluation, we report both standard mAP in all individual classes (Everingham et al. 2010) and the inference speed of the detectors. A detection is considered to be correct when the **UnbIoU** between the prediction and ground-truth exceeds 50%, which is unbiased and fair for all comparison methods.

Rep R-CNN on Different Criteria

To give a convincing result among criteria, we exploit different criteria in the Rep R-CNN and transform the part originally proposed for spherical criteria to the compared criteria. Specifically, for ERPBB and CirBB, we project the objects on the sphere back to ERP, and utilize rectangles and circles to tightly bound the objects. Regarding TanBB, we sample 24 points on the outline of each BB for PolyIoU computing. The SphAnchor and spherical bounding-box regression are both adjusted to the corresponding criteria. For a fair comparison, we use the same network architecture in Rep R-CNN and fine-tune the parameters for different criteria.

As shown in Table 1, the proposed spherical criteria outperforms the other existing criteria under the similar inference time, and achieves highly competitive results compared with unbiased criteria on both VOC360 and COCO-Men datasets while running at 3x speed. Though TanBB is also an unbiased measurement of UnbBB, the use of PolyIoU degrades the performance in both mAP and speed. Besides, since ERP-based criteria introduce bias in BBs, they get much lower mAP than the other criteria.

Performance of Two-stage Rep R-CNN

Baseline Methods: We take the following state-of-the-art networks as the baseline methods.

	n	VOC360	COCO-Men	Speed
Multi-projection YOLO	200	54.29	61.02	273ms
Sphere-SSD	-	48.25	54.79	86ms
SPHCNN	-	49.41	48.17	224ms
S^2CNN	-	37.45	45.36	139ms
Spherical CNN	-	35.12	41.53	145ms
SpherePHD	-	50.79	59.69	323ms
G-SCNN	50	50.30	56.03	202ms
Multi-kernel	50	44.23	51.99	144ms
Rep R-CNN	10	69.70	65.43	112ms
Rep R-CNN	20	**71.88**	74.72	127ms
Rep R-CNN	50	71.65	**81.48**	178ms
Rep R-CNN	100	71.57	80.34	256ms

Table 2: Performance comparison between baseline methods and Rep R-CNN on both VOC360 and COCO-Men datasets. The boldface denotes the best performance on each dataset.

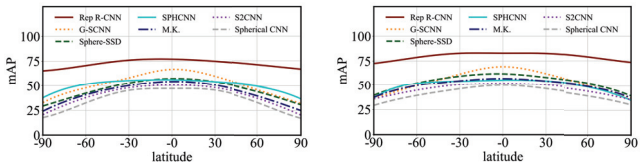

Figure 5: Latitude/mAP curves of Rep R-CNN and baseline methods on (a) VOC360 and (b) COCO-Men.

(1) Multi-projection YOLO (Yang et al. 2018): The overlapping PSPs are selected to cover the sphere, and then fed to the YOLO detector (Redmon and Farhadi 2017).

(2) Sphere-SSD (Coors, Paul Condurache, and Geiger 2018): The Sphere-SSD simply replaces normal convolutions in the vanilla SSD by SphConv.

(3) SPHCNN (Su and Grauman 2017): We add a 3×3 conv on top of conv5_3 feature map of SPHCNN (Su and Grauman 2017) for bounding-box regression and classification in real-world 360° detection scenario.

(4) S^2CNN (Cohen et al. 2018): We follow the authors' implementation of S^2CNN, and adapt it to the detection task by adding a 3×3 conv on top of the combined feature map. To avoid out-of-memory error, we scale down the input resolution to 64×64 as suggested by authors.

(5) Spherical CNN (Esteves et al. 2018): We modify the authors' implementation in the same way as S^2CNN. The input is again scaled down to 64×64 due to the memory limit.

(6) SpherePHD (Lee et al. 2019): We follow the authors' open source CNN which takes icosahedral mesh as input and apply YOLO detector (Redmon et al. 2016) for 360° detection.

(7) G-SCNN (Yu and Ji 2019): We refer to the two-stage G-SCNN which transforms the first-stage ERP to grid map and applies both S2 and SO(3) conv on it in the second stage.

(8) Multi-kernel (Wang and Lai 2019): We utilize the multi-kernel layers and position information in a Faster R-CNN.

All the baseline methods except Multi-projection YOLO and SpherePHD take ERP as original input, and the backbone networks are all the same VGG-16 architecture. Besides, since most of the existing methods utilize the biased ERPBB and ERPIoU as criteria, if we only take the original implementation of those methods, they will definitely

(a) VOC360 (b) COCO-Men (c) SUN360

Figure 6: Detection results of Rep R-CNN on the three datasets.

get poor results. Hence, we adopt the proposed **spherical criteria** in all one-stage baseline methods which only affects the final detection procedure. For the two-stage detectors, we only apply spherical criteria in the second stage. The main consideration is that all two-stage baselines apply RoI pooling directly on the extracted feature map. If we use spherical criteria in the first stage, the proposals of SphBBs would correspond to distorted regions on ERP, which can not be appropriately transformed to fixed-size features on ERP. Thus, we reserve the original ERP-based criteria in the first stage. In addition, all methods are tuned by either implementation with recommended parameters or grid search for the best performance. Please refer to supplementary material for additional details of datasets and baseline methods.

Comparison with Baseline Methods: We compare Rep R-CNN with the baseline methods in both VOC360 and COCO-Men datasets. The results are shown in Table 2. It is obvious that Multi-projection YOLO achieves the best performance in the baseline methods owing to the undistorted PSP, but is also time-consuming for the large number of samplings. SpherePHD also performs well in both datasets, but suffers from the oversampling in icosahedron and results in low speed. Based on the two-stage framework, G-SCNN and Multi-kernel get higher mAP than one-stage S^2CNN and spherical CNN, but are still inferior to the above methods due to the biased ERP feature generated from the first stage. Among those one-stage detectors, Sphere-SSD exhibits competitive performance in both datasets with almost 3x speed faster than both Multi-projection YOLO.

Regarding Rep R-CNN, it can be observed that the proposed detector adopts the rapid and relatively precise SphConv in SphRPN, and then regresses the predictions with the accurate PSP-based RepNet. Therefore, Rep R-CNN combines the advantages of both Multi-projection YOLO and Sphere-SSD, which is fast and accurate. The results in VOC360 and COCO-Men convincingly demonstrate the effectiveness of the proposed method. Specifically, Rep R-CNN achieves 71.88 mAP on the VOC360 dataset and 81.48 mAP on the COCO-Men dataset, exceeding the strongest baseline, *i.e.*, Multi-projection YOLO, by over 30% in both datasets. In addition, Rep R-CNN achieves the best performance with competitive speed , which is faster than almost all the baseline methods.

Moreover, to show the robustness of Rep R-CNN, we examine the mAP of the detection algorithms by varying the latitude, and plot the latitude/mAP curves in Figure 5. We only consider the methods that are affected by latitude (take ERP as input). It is obvious that Rep R-CNN forms an upper envelope over all existing methods. Furthermore, though the distortion in ERP varies with latitude, Rep R-CNN is only slightly affected, and exhibits competitive performance even if the objects are extremely distorted, *i.e.*, near the poles. Some predictions of Rep R-CNN are visualized in Figure 6. The result reveals that Rep R-CNN is robust to the various distortion and discontinuity situations.

Rep R-CNN over Real-world Dataset

To verify that the proposed Rep R-CNN is also effective in actual scenario, we exploit the Rep R-CNN trained on VOC360, and directly apply it to the real-world SUN360 dataset. As shown in Figure 6.c), consistent with the previous experiments, objects of various categories at different latitudes can be successfully detected despite the distortion and discontinuity, demonstrating that Rep R-CNN could perform well over the real-world scenarios. Additional qualitative results are provided in the supplementary material.

6 Conclusion

In this paper, we present a fast and accurate spherical criteria, including SphBB and SphIoU, for 360° object detection, and introduce a novel two-stage object detector named Rep R-CNN by combining the strength of both ERP and PSP. Experimental results on the novel synthetic VOC360 and COCO-Men datasets show that both spherical criteria and Rep R-CNN outperforms several state-of-the-art 360° object detectors with the similar computation overhead, verifying efficiency and accuracy of the design. In addition, the model can be transferred to the real-world SUN360 dataset and remain good detection performance, indicating that Rep R-CNN is applicable to the real-world scenarios.

Acknowledgement

This work is partially supported by the National Key Research and Development Program No. 2017YFB0803302 and the National Natural Science Foundation of China under Grant No. 61572051.

References

Ardouin, J.; Lécuyer, A.; Marchal, M.; Riant, C.; and Marchand, E. 2012. Flyviz: a novel display device to provide humans with 360 vision by coupling catadioptric camera with hmd. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, 41–44.

Boomsma, W., and Frellsen, J. 2017. Spherical convolutions and their application in molecular modelling. In *Advances in Neural Information Processing Systems*, 3433–3443.

Cheng, H.-T.; Chao, C.-H.; Dong, J.-D.; Wen, H.-K.; Liu, T.-L.; and Sun, M. 2018. Cube padding for weakly-supervised saliency prediction in 360° videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1420–1429.

Cohen, T. S.; Geiger, M.; Köhler, J.; and Welling, M. 2018. Spherical cnns. In *ICLR*.

Cohen, T.; Weiler, M.; Kicanaoglu, B.; and Welling, M. 2019. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, 1321–1330.

Coors, B.; Paul Condurache, A.; and Geiger, A. 2018. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 518–533.

Coxeter, H. S. M. 1961. Introduction to geometry.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. *CoRR, abs/1703.06211* 1(2):3.

Esteves, C.; Allen-Blanchette, C.; Makadia, A.; and Daniilidis, K. 2018. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–68.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88(2):303–338.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Huang, J.; Chen, Z.; Ceylan, D.; and Jin, H. 2017. 6-dof vr videos with a single 360-camera. In *2017 IEEE Virtual Reality (VR)*, 37–44.

Jiang, C.; Huang, J.; Kashinath, K.; Marcus, P.; Niessner, M.; et al. 2019. Spherical cnns on unstructured grids.

Kern, W. F., and Bland, J. R. 1938. *Solid mensuration: with proofs*. J. Wiley & Sons, Incorporated.

Khasanova, R., and Frossard, P. 2017. Graph-based classification of omnidirectional images. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 860–869.

Lee, Y.; Jeong, J.; Yun, J.; Cho, W.; and Yoon, K.-J. 2019. Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9181–9189.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37.

Preparata, F. P., and Shamos, M. I. 2012. *Computational geometry: an introduction*. Springer Science & Business Media.

Redmon, J., and Farhadi, A. 2017. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. M. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3234–3243.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115(3):211–252.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Snyder, J. P. 1987. *Map projections–A working manual*, volume 1395. US Government Printing Office.

Snyder, J. P. 1997. *Flattening the earth: two thousand years of map projections*. University of Chicago Press.

Su, Y.-C., and Grauman, K. 2017. Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, 529–539.

Su, Y.-C., and Grauman, K. 2019. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9442–9451.

Tateno, K.; Navab, N.; and Tombari, F. 2018. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 707–722.

Wang, K.-H., and Lai, S.-H. 2019. Object detection in curved space for 360-degree camera. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3642–3646. IEEE.

Weisstein, E. W. 2003. Circle-circle intersection.

Xiao, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2012. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2695–2702.

Yang, W.; Qian, Y.; Cricri, F.; Fan, L.; and Kamarainen, J.-K. 2018. Object detection in equirectangular panorama. *arXiv preprint arXiv:1805.08009*.

Yu, D., and Ji, S. 2019. Grid based spherical cnn for object detection from panoramic images. *Sensors* 19(11):2622.

Zhao, Q.; Zhu, C.; Dai, F.; Ma, Y.; Jin, G.; and Zhang, Y. 2018. Distortion-aware cnns for spherical images. In *International Joint Conferences on Artificial Intelligence*, 1198–1204.