Six-to-one: Cubemap-guided Feature Calibration for Panorama Object Detection

Jingbo Miao*†, Yanwei Liu*‡, Kan Wang*†, Jinxia Liu§, Antonios Argyriou¶, Yanni Han* and Zhen Xu*
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
§Zhejiang Wanli University, Ningbo, China
¶University of Thessaly, Volos, Greece
{miaojingbo, liuyanwei, wangkan, hanyanni, xuzhen}@iie.ac.cn, liujinxia1969@126.com, anargyr@uth.gr

***Abstract*—Object detection methods for perspective images have proven increasingly efficient, but the techniques for equirectangular projection (ERP) panoramas from inherently spherical imaging cannot still achieve satisfactory performance. Due to the various degrees of distortion at different pixel locations, current algorithms cannot adapt to the changes in shape and contour caused by stretching, which results in performance degradation when migrating them from perspective images to spherical ones. In this paper, we improve the network for panorama object detection and introduce the cube-domain information with discontinuity but low distortion to correct the panorama features. Unlike previous works, we consider the impact of semantic discontinuity from all tangent planes instead of overlaying features when needed. Considering the six facets as unified, i.e., six-to-one for extraction, the proposed Facet-Link module enhances the long-range sensing capability at the facet level in the frequency domain. Moreover, the position alignment packs different facets, i.e., six-to-one for calibration, to preserve more global signals during the correction stage, which establishes semantic pathways for feature interactions between panorama and cubemap in the two dimensions, facet-facet and cube-pano, respectively. Extensive experiments on synthetic and real-world datasets verify the effectiveness and robustness of our proposed method.**

***Index Terms*—panorama, feature calibration, cubemap, object detection**

I. Introduction

Object detection for flat images has undergone a long development. The introduction of the Region with Convolutional Neural Network (RCNN) [1] pioneered the two-stage framework. Then the emergence of YOLO [2] puts the end-to-end detection framework on the runway by splitting the image into multiple patches and pre-setting the anchor which is optimized in the loss function for prediction. The original target of these neural networks is traditional perspective images. The “proposal” of two-stage networks and the “anchor” of single-stage networks are both designed under the rule of pinhole imaging. Thus, these previous methods have limited applicability for the images from spherical imaging patterns.

Spherical images provide a wider field-of-view (FoV) than traditional images, thus accommodating more spatial and semantic information. They have played an important role in application scenarios such as autonomous driving [3] and mixed reality [4]. Due to the images being spherical in origin, the distortion of planar format is obviously severe, and the semantic representation is not intuitive. Therefore, directly applying existing networks limits the performance, especially in high latitude regions. One approach [5] provides convolution filters with different shapes for different levels of distortion. However, this method is similar to manual configurations and insufficient to capture position-based distortion characteristics. Other methods directly transform the convolutional kernel on the sphere [6], SO(3) domain [7] , and icosahedral mesh [8], which achieve a more suitable feature extraction by converting the planar format to these alternative expressions. But these methods, usually in the spherical domain or an even higher dimensional space, suffer from greater computational volume and usage costs that increase with the resolution of images. As an alternative format of spherical images, the cubemap [9] represents features of different perspectives in sub-maps with fewer distortion, which is undoubtedly a booster for improving the performance of panorama networks. Bifuse [10] fuses features from the cube-domain and pano-domain to predict depth. However, it is not sensible to simply superimpose features to contribute different views to the pano-domain with the same weight because cubemap and panorama undergo utterly different projection patterns. The spherical padding only “physically connects” different facets, and the semantic information cannot effectively interact.

In this paper, we optimize an end-to-end object detection framework suitable for 360° images, which is based on introducing cubemap format together with panorama as inputs to the network. We extract the global features in the frequency domain to deal with the applicable position embeddings among multiple facets and also capture the spatial domain features to track the fine-grained and short-range signals by the low distortion within the facets. Not only physically but also semantically, we align and pack the six facets into one feature unit to rectify the panorama stream by structuring attention at the facet level.

The main contributions of this paper are summarized as follows:

- The proposed feature calibration for end-to-end panorama object detection, uniting six facets of cubemap into one

‡Corresponding author. This work was supported in part by National Natural Science Foundation of China under Grant 61771469 and the Cooperation project between Chongqing Municipal undergraduate universities and institutes affiliated to CAS (HZ2021015).

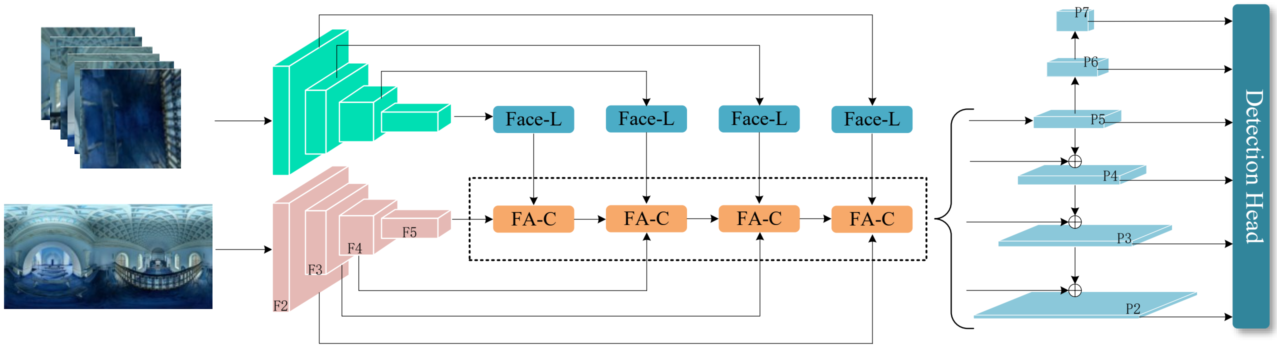


Fig. 1. The proposed framework. The network takes cubemap and panorama as inputs to predict categories and bounding boxes. The cubemaps of each layer link six facets to offset the discontinuity (”Facet-L”) and are packed to calibrate the corresponding panorama (”FA-C”). Then the corrected features participate in the pyramid and head network.

semantically, attenuates the side effects of distortion to improve the performance of the network.

- We utilize both the global and local features from the cubemap stream. Placing the six facets in the same frequency range involves all features in extraction, which makes up for the discontinuities from different facets.
- We embed a facet-aware calibration module into the multi-scale information fusion flow. We aggregate and pack facets for position registration and then propose the FACA operator for adaptive feature correction guided by the cubemap domain.

II. Related Work

A. Object Detection for Perspective Images

Object detection technology for perspective images has been gradually improving. The convenience brought by CNNs constitutes the two-stage network represented by RCNN [1] and the single-stage network represented by YOLO [2] good foundations for achieving impressive accuracy and speed. As the mainstream backbone networks, the residual network [11] and the spatial channel separable Inception-net [12] have become the primary modality for feature extraction in a number of studies and have shown promising performance and generalizability. At the neck layer, the feature pyramid network [13] bridges the gap between different convolutional layers of the backbone network and becomes the beginning of multi-scale detection techniques. The advent of CenterNET [14] frees the network from the constraints of anchors. To break the limitations of local perception of CNNs, Wang et al. [15] adopts a self-attention mechanism to unlock the long-distance dependence of the network. SENet [16] and CBAM [17] employ channel and spatial attention to capture the importance of each partition. The low complexity of traditional convolutional networks is well suited for vision tasks. The rich selection of backbone, multi-scale prediction, and attention mechanisms are essential guides for the detection task of the panorama. However, due to the non-negligible distortion, these methods require a series of optimizations to perform better for spherical images.

B. Object Detection for Panorama

The most straightforward approach to combating distortion is to directly correct the distorted signal for the following downstream tasks, as [18] investigates stereo correction strategies using 3D geometry. However, correction in the 2D plane inevitably loses the information initially contained in the image, which naturally affects the effectiveness of object detection. Detecting panorama objects by 2D convolution, Su et al. [5] modify the shape of the convolution kernel along the meridional direction for panoramas. Yang et al. [19] project the image to multiple views for object detection separately, which is regarded as a rudimentary but effective approach. Spherenet [6] projects the filtered region onto the tangent plane to adapt the filter to a specific position. Unlike 2D conventional convolutional networks, based on the isomorphism between the rotation of the sphere in 3D space and the 3D rotation group SO(3), Cohen et al. [7] convert the image from the 2D plane to the SO(3) domain to perform the correlation operation on the hypersphere space. SpherePHD [8] applies CNNs to icosahedral mesh and proposes triangular filters that take into account orientation with their corresponding pooling layers. Recent studies have started to develop suitable evaluation metrics [20] and detect objects [21] based on spherical coordinates. Still, extending these methods to more datasets and corresponding applications is problematic because getting more spherically labeled data for training in real scenarios is costly. As multiple information branches, 3D target detection and depth are used to strengthen the semantic perception of the network in [22]. Bifuse [10] improves the accuracy of panorama depth estimation by means of a cubemap branch with the benefit of low distortion and edge padding. However, their feature extraction for cubemap does not consider the semantic connection between the facets, and each facet of the cubemap has the same influence factor on the equirectangular projection (ERP) panorama branch, which will be discounted in detection performance.

III. Method

In this section, we detail the object detection architecture for panorama. We first introduce the general framework, followed by Facet-Link Block and facet-aware calibration in order.

A. Framework

We show the basic framework of the feature extraction in Fig. 1. The initial stage consists of two branches guided by panorama and cubemap, respectively. The final head network deduces the results based on the corrected panorama stream. The feature extraction mainly includes four building blocks: (1) backbone network stream of panorama; (2) backbone network stream of cubemap; (3) Facet-Link block that unifies the facets of cube-domain; (4) Facet-aware calibration that packages individual facets and performs weighted fusion with panorama. The panorama and cubemap branches use the classic ResNet [11] to extract the raw signals (including edges, contours, etc.) of the image and express them in a high-dimensional space.

Since objects in the panorama are usually stretched, especially at high latitudes, a more flexible prediction scale leads to better performance. FPN [13] enables the network to predict at multiple scales. It opens up the information transmission paths of the backbone network at different stages and combines the high-level rich semantics with all other scales by a top-down architecture. We build the pyramid structure by extracting each feature map $\mathrm{f_l} \in \{res2/_2, res3/_3, res4/_5, res5/_2\}$ of the ResNet-101, and their stride gets progressively larger as $\{4, 8, 16, 32\}$. The generated corresponding pyramid structure of the extracted feature map $\{F_2, F_3, F_4, F_5\}$ builds pathways to enrich semantic information.

Unlike the original design in FPN, the panorama stream of our network, followed by the skip-connection structure, is rectified by cubemap with lower distortion. To counter the inherent discontinuity among facets, the Facet-Link block, abbreviated as ∇_L, links the features of the cubemap as a whole. It captures the position dependence over long distances and enhances the local features within the frequency and spatial domains, respectively. Furthermore, we fully consider the position alignment of the multiple facets of cubemap with the single one of panorama in the facet-aware calibration block ∇_C, making the features adaptable to different positions for the following proposed calibration operator FACA. We formalize the pattern of these components being embedded in the network as:

$$fe_i^{'} = \nabla_C(fe_i, \nabla_L(fc_i)) \tag{1}$$

where fc_i and fe_i denote feature maps in cube stream and panorama stream, respectively. Note that these proposals are structured in the multi-layers pyramid, and the panorama feature maps are corrected in different scales. Mathematically, the pyramid fusion is represented in the form of:

$$fe_i^{''} = fe_i^{'} \cdot \delta(fe_{i+1}; \epsilon) + fe_i^{'} \tag{2}$$

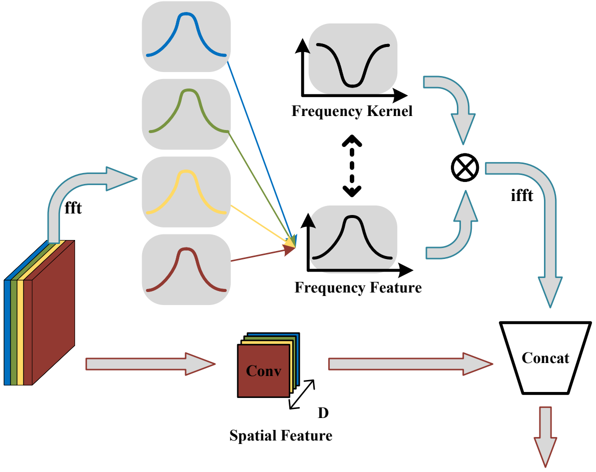


Fig. 2. Facet-Link Block. Signals are extracted by conventional 2D convolution in the local stream. In the global stream, FFT transforms them into a spectrum with learnable parameters for long-range perception.

where fe_{i+1} denotes the previous layer of fe_i in the panorama stream with a larger size. $\delta(\cdot)$ denotes 1×1 convolution operation and ϵ is the corresponding parameters.

The objects on panoramas, especially at high-latitude positions, are usually wider than those on traditional flat images. To get a larger receptive field for larger bounding boxes, we extend P_5 for the two deeper layers $\{P_6, P_7\}$ to get the final features $\{P_2, P_3, P_4, P_5, P_6, P_7\}$ for the subsequent steps of anchor selection and detection head network.

The cubemap branch also uses ResNet-101 for the initial feature extraction, which acts more like an auxiliary reinforcement branch, so it does not use the top-down pathway for feature enhancement. To maintain the corresponding coordination with the panorama stream in size, we also extract layers ranging from 2 to 5 as the ingredients of feature calibration.

B. Facet-Link Block

The cube stream is only limited to simple feature extraction in the former panorama image processing methods enhanced by cubemap. Cubemap can complement panorama feature extraction due to its less distortion within each facet. The downside is that the six disconnected facets cause signal boundary discontinuities and impair the performance of feature extraction networks. So our method treats the six facets as a whole, i.e., six-to-one, to capture facet-to-facet positional dependencies and make the information semantically related to each other.

To maximize the benefits of low distortion, we propose a cube-domain stream that integrates different facets based on both global and local branches, as shown in Fig. 2. Mathematically, we formalize the output from the backbone network as the cube features $\mathcal{CM}_i \in \mathbb{R}^{H \times W \times C_f}$, where H and W represent the length and width of the tangent maps. C_f denotes the number of channels included in each facet i,

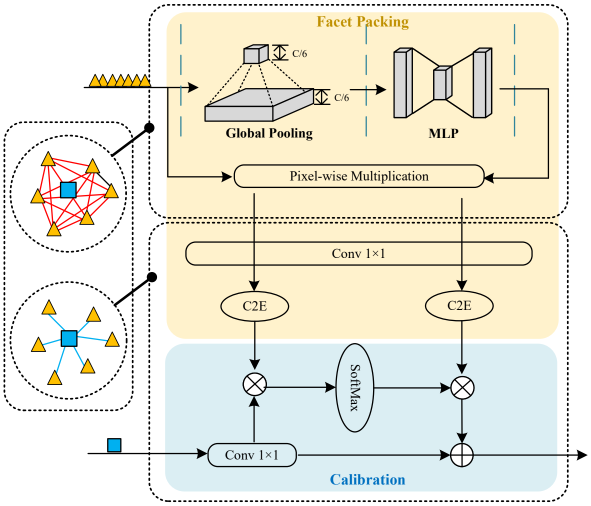


Fig. 3. Cubemaps weight and pack the features at the facet level (shown in the yellow region) to establish position alignment; the panorama is corrected on a facet-by-facet scale (shown in the blue region). The dashed circle indicates the association between different parts of the feature maps, where the yellow triangles indicate the facets of the cubemap and the blue square represents the panorama.

$i \in \{B, D, F, L, R, U\}$ representing the six directions of the spherical projection, back, down, front, left, right, and up. We feed it into the local branch for fine-grained spatial perception and note that "local" here indicates feature extraction within the channel. Since planar CNNs can share weights among various kernels and the feature map presents local sensory fields, the conventional convolution can be competent for this task to get $\mathcal{CM}_{i\_local}$. In the global stream, we transfer the cubemap to the frequency domain through 2D Discrete Fourier Transform (DFT) Ψ for pixel $[k, l]$,

$$\Psi[k,l]=\frac{1}{HW}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1} f[h,w]e^{-j2\pi(\frac{k}{H}h+\frac{l}{W}w)} \tag{3}$$

where $0 \leq h \leq H-1$ and $0 \leq w \leq W-1$. In practice, we adopt the Fast Fourier Transform (FFT) by decomposing the DFT matrix into the product of sparse (mostly zero) factors to speed up the calculation. It reduces the complexity of calculating the DFT from $O(N^2)$ required to calculate only the DFT definition to $O(N \log N)$, depending on the image size N. As demonstrated by Rao et al. [23], the frequency domain convolution operation is equivalent to a global round-trip convolution between channels. By constructing a frequency domain convolution kernel of the same size $W_i \in \mathbb{R}^{H\times W\times C_f}$, we will finish cubemap feature extraction in the spectrum:

$$\widetilde{\mathcal{CM}_i} = W_i \circ \Psi(\mathcal{CM}_i), i \in \{B, D, F, L, R, U\} \tag{4}$$

where $\circ$ represents the entry-wise product (also known as the element-wise product). After rendered to the frequency domain, the facets from the perspective domain of different projection angles enjoy the same frequency interval. That is, the domain switching aligns the six facets i to be extracted for semantic signals to the same scale. And using different parameters W_i separately ensures the weight sharing of the convolution, which guarantees to retain the uniformity among the features with a reduced number of parameters.

The frequency-domain cubemap features are then converted back to the spatial domain by the inverse Fast Fourier Transform (IFFT), and the ones from the local stream are concatenated with them:

$$\mathcal{CM}_{i\_link} = \mathcal{F}[\mathcal{CM}_{i\_local} \cdot \Psi^{-1}(\widetilde{\mathcal{CM}_i})] \tag{5}$$

In the above, the function $\mathcal{F}[\cdot]$ indicates that the feature maps are concatenated in the channel direction followed by a 1×1 convolution layer. It should be mentioned that the final convolution reduces the number of channels to be consistent with the ones in the panorama stream.

C. Facet-aware Calibration

In this section, we detail feature alignment and packing for facets, i.e., six-to-one, at the semantic level, followed by a calibration operator. As shown in Fig. 3, we develop semantic links between facet and facet as well as that between panorama and cubemap.

The cubemap format has multiply-disconnected but low distorted facets rendering different tangent planes separately. When invested into the panorama branch as corrective signals, these facets equally affect all positions of the target features by regular addition or multiplication, which does not correspond to the actual situation. Our proposed calibration module encodes the features from Facet-Link at the facet level so that each rectified feature can be adaptive to the specific position.

We use the most classic channel substructure [16] for the position registration. First, the feature map undergoes a global pooling F_{gp} operation. It is worth emphasizing that to highlight the characteristics within the individual facets of cubemap, we split the channel into facet units (6 partitions) and pool all channels C_f involved in each portion. More strictly, the pooled pixel value ${p'}_c \in \mathbb{R}^{H\times W\times 6}$ is the average result derived by compressing the previous feature layer in units of size $H \times W \times C_f$:

$${p'}_c = F_{gp}(p_c) = \frac{1}{H\times W\times C_f}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{C_f} p_c(i,j,k) \tag{6}$$

We achieve the mining of facet dependencies by employing facet-level pooling, and the results are fed as descriptors into a Multi-layer Perception (MLP) layer. We formalize the gating mechanism as follows:

$$\mathcal{G} = \sigma(W_p F_{gp}(\mathcal{CM}_{i\_link})) \tag{7}$$

where $W_p \in \mathbb{R}^{C_f/r\times C_f}$, r is the reduction ratio and σ denotes the sigmoid function. Then we weight the facets by element-wise multiplication,

$$\begin{gathered}\mathcal{CM} = \sum_{i=1}^{6} \mathcal{G} \otimes \mathcal{CM}_{i\_link} \\ i \in \{B, D, F, L, R, U\}\end{gathered} \tag{8}$$

TABLE I

Performance comparison on VOC-360 and 360-Indoor datasets. (P/S) denotes the source network designed for perspective or spherical images. (*) denotes the detection results are derived from spherical coordinates. The bold numbers represent the best performance in each case.

Method	Backbone	Type (P/S)	360-Indoor			VOC-360		
			AP (%)	AP_{50} (%)	AP_{75} (%)	AP (%)	AP_{50} (%)	AP_{75} (%)
Faster-RCNN [26]	ResNet-101	P	18.3	38.7	14.1	55.2	65.7	49.8
CenterNET [14]	ResNet-101	P	18.0	37.9	13.3	58.5	**73.5**	53.6
YOLOv5 [2]	DarkNet-53	P	17.9	38.4	13.4	49.2	59.8	42.9
S2CNN [7]	ResNet-101	S	10.0	25.7	7.9	26.1	28.1	20.3
Rep-RCNN [21]	ResNet-101	S	5.0*	15.3*	1.9*	-	-	-
Dai et al. [20]	ResNet-101	S	10.0*	24.8*	6.0*	-	-	-
SphereNet [6]	ResNet-101	S	16.7	34.3	13.8	47.9	52.1	48.7
OURS	ResNet-101	S	**21.5**	**40.2**	**19.6**	**62.8**	67.9	**62.7**

Furthermore, the ERP single-sided format needs to interact with the multi-sided information in cubemap to get rectified by the lower-distortion features without losing the overall image signals. As shown in the blue area of Fig. 3, we exploit the soft addressing property of cross attention to make the fragmented but less distorted semantic information from the cube-domain correct the global features $\mathcal{E} \in H \times W \times C$ in the panorama stream. The specific cube facet adaptively finds the most closely related regions for feature correction and ignores the unrelated panorama features. To this end, we propose facet-aware cross-attention (FACA) as follows:

$$FACA(\mathcal{E}) = Softmax(\Gamma(\delta(\mathcal{CM}))\otimes\delta(\mathcal{E}))\otimes\Gamma(\delta(\mathcal{CM})) \quad (9)$$

where $\delta(\cdot)$ denotes a 1×1 single layer convolution and the softmax function normalizes the attention score. $\Gamma(\cdot)$ is the C2E function that converts cubemap to equirectangular projection format.

We render the texture to the sphere by orienting the coordinates of the cubemap from {B,D,F,L,R,U} six views by inverse mapping function:

$$q_{si} = R_{f_i} \cdot q_{ci} \quad (10)$$

where R_{f_i} is the rotation matrix in the coordinate system to the center of the cube, and q_{ci} and q_{si} are the corresponding points from the tangent plane and sphere, respectively.

Given the longitude ϕ and latitude θ, where $0 \leq \phi \leq 2\pi$ and $0 \leq \theta \leq \pi$, we calculate the average of several nearest neighboring pixels to identify which facet of the cubemap that $q_{si} \in [\phi, \theta]$ lies on and the specific pixel value of the facet. Then we get the position (X, Y, Z) on the equirectangular coordinate by mapping the spherical position (ϕ,θ) to the panorama domain:

$$\begin{aligned} X &= cos(\theta)cos(\phi) \\ Y &= sin(\theta) \\ Z &= -cos(\theta)sin(\phi) \end{aligned} \quad (11)$$

The schematic diagram at the left of Fig. 3 shows that our proposed calibration strategy considers the connectivity of different facets of the cubemap and their correlation with the overall signals (ERP format) at the two successive stages, which performs adaptive position registration among the facets under the spherical global scope for the panorama rectification. Therefore, we compensate for the negative impact of border discontinuities to enhance the features semantically for the downstream task.

IV. Experiments

In this section, we evaluate the proposed method by comparing it with the advanced object detection networks; then show the ablation study results.

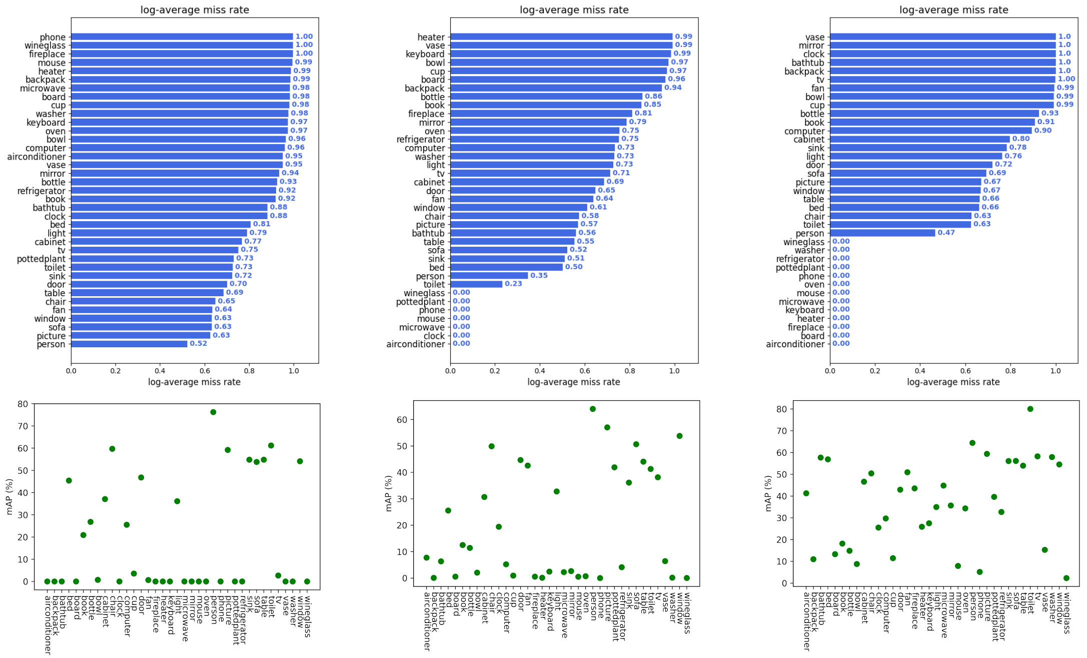
Datasets. To demonstrate the performance and generalization ability of the model more comprehensively, we adopt both real-world and synthetic datasets.

360-Indoor [24]: An open-source panorama object detection dataset. It collects and annotates objects in complex indoor scenes with 37 categories, consisting of 3k images and the corresponding total of 90k annotations. After projecting the panorama onto the spherical tangent plane, we use mapping functions to convert the bounding FoV from a spherical rectangle (θ, ϕ, α, β) to the regular bounding box (x_{min},y_{min},x_{max},y_{max}) as input to the network.

VOC-360: A synthetic dataset based on the PASCAL VOC 2012 [25] with 20 classes. We randomly project the perspective images onto the sphere along the longitudinal direction as {0°, 36°,54°,72°} and subsequently expand them into panorama as the input to the network. Each image is attached to only one object, and the regions outside the target are filled with zero values. 18k training images, 6k validation images, and 3k test images are available in VOC-360.

Experimental Settings. The network takes 1920×960 as the size of the input, including the training and testing stage. We use stochastic gradient descent (SDG) as the optimization method with momentum set to 0.9. The learning rate is initialized as $5e^{-2}$ and gradually decays by e^{-4} to update the weights on both 360-indoor and VOC-360. The network is trained on two NVIDIA Tesla P40 (24G) GPUs with the batch size of 2×8.

Augmentation. Since different positions of equirectangular projection suffer from various degrees of distortion, stitching across longitudes on the panorama format causes inconsistency



(a) SphereNet (b) Faster-RCNN (c) Ours

Fig. 4. The performance comparison in category granularity. The bar graph shows the log-average miss rate (top), and the scatter plot shows the mAP distribution (bottom) of each object category.

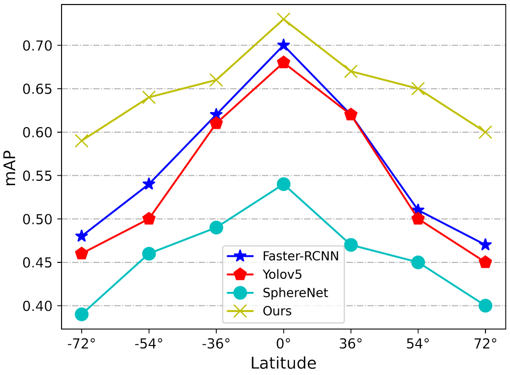


Fig. 5. The mAP comparison of each network at different latitudes and their respective trends on the VOC-360 dataset.

between distortion and patch. To improve the generalization ability of our model, we rotate the panorama along the yaw axis by 90°. For indoor datasets, each category has a different distribution in different latitudes, and radical changes may affect the realism of the dataset. Therefore, we do not change the polar position of the objects.

A. Performance Comparison

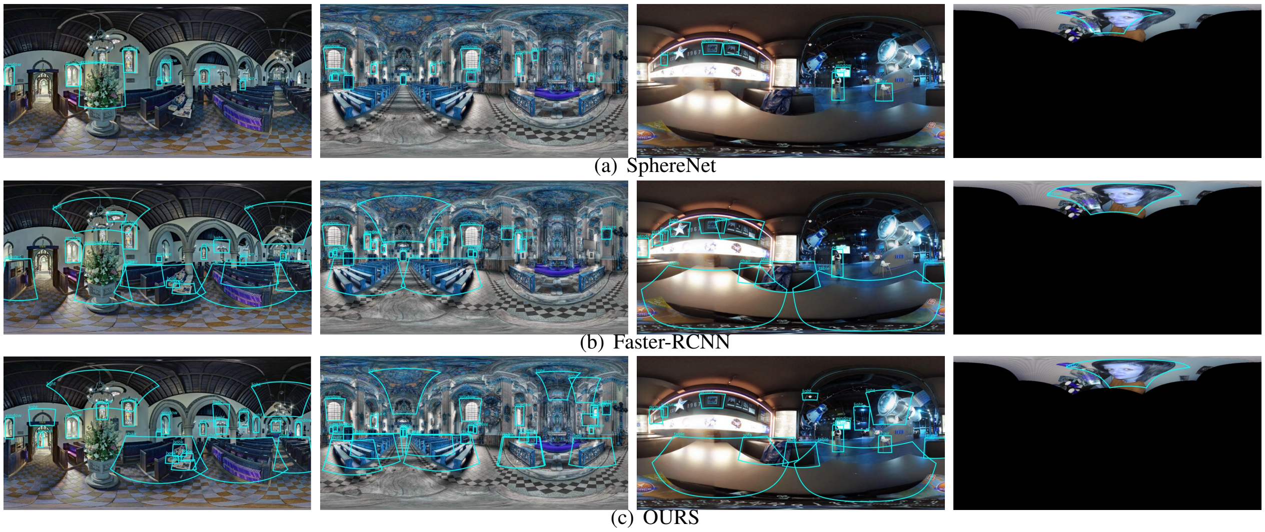
Baselines. We compare the accuracy of the proposed network with the traditional planar and spherical methods.

Planar methods. We chose Faster-RCNN [26], YOLOv5 [2], and CenterNET [14] as they represent networks for the three most classic planar detection architectures: two-stage, one-stage, and anchor-free. Objects in panorama have a larger size and different aspect ratio, so we modify the size of the anchor box to be 2.3 times larger than the COCO dataset in all networks. We use ResNet-101 pre-trained on COCO as the backbone to get better results. Since the YOLO model with ResNet suffers a poor performance, we choose the more suitable Darknet-53 as an alternative.

Spherical methods. We choose SphereNet [6], S2CNN [7], Rep-RCNN [21], and Dai et al. [20] since they are both "tailor-made" methods for spherical images. Regarding the methods based on spherical coordinates that differ from ours in terms of detection rules, e.g., IOU calculation, we just include the detection results of Rep-RCNN and Zhao et al [20]. since their codes are not publicly available. S2CNN maps images to the SO(3) domain and a complete application in the backbone will inevitably lead to a severe memory crisis, so we only replace the 2D convolution with S2CNN for the last two layers.

Metrics. We use the same evaluation metrics as MS COCO [27] to obtain AP@[.5:.05:.95] with IoU threshold ranging from 0.5 to 0.95 with the step size of 0.05.

The performance comparison is presented in Table I. It can be seen that in both datasets, our proposed network performs better. In the real-world 360-indoor dataset, our method outperforms the Faster-RCNN by 3.2% in terms of mAP. For the same spherical methods, the proposed method is four percentage points higher than SphereNet. The results on VOC-360 have an overall higher accuracy than those on the real-world one since each image of VOC-360 contains only a single target. Our model is still 9.1% higher than CenterNET in AP_{75}. Probably due to the limitation of factors resulting in a lower benchmark, such as the bounding box settings, spherical



(a) SphereNet

(b) Faster-RCNN

(c) OURS

Fig. 6. Examples of detected results on 360-Indoor and VOC-360.

TABLE II
ABLATION STUDIES ON 360-INDOOR DATASETS. "F-L" AND "CAL" INDICATE FACET-LINK AND CALIBRATION MODULES, RESPECTIVELY.

Input	Backbone	+F-L	+Cal	360-Indoor AP	AP_{50}	AP_{75}
Pano	ResNet-50			9.7	23.2	7.1
Pano+Cube	ResNet-50			12.1	27.9	9.8
	ResNet-50	✓		14.3	30.1	10.6
	ResNet-50	✓	✓	16.6	30.5	12.5
Pano	ResNet-101			11.4	24.0	10.9
Pano+Cube	ResNet-101			13.7	27.7	10.4
	ResNet-101	✓		16.3	31.2	12.0
	ResNet-101	✓	✓	17.1	32.5	13.9

IOU calculation, etc., the performance of the methods (Rep-RCNN and Dai et al. [20]) based on spherical coordinates is relatively poor.

Compared to the planar methods, our AP_{75} metric has a higher value because the top facet from the cube domain provides more cues to the model's inference semantically, and the model gives more confidence for highly distorted objects. CenterNET shows the highest value of AP_{50} on the VOC-360 dataset. Probably not being restricted by anchors, it can outperform other networks at lower confidence. In addition, we also notice its instability in the overall performance.

We also compared the accuracy trends among the networks on VOC-360 at different latitudes, as shown in Fig. 5. The accuracy of the spherical model is more uniform at different latitudes. In general, our model has the highest accuracy in seven latitudes among all methods. On the contrary, the planar network performs well near the equator but deteriorates sharply close to the high-latitude positions owing to the penalty of image distortion, which also implies the robustness and stability of our model on different datasets.

Fig. 4 shows the log-average miss rate of different networks in each category, and our proposed network has lower error rates (even to zero). Note that our method performs worse in some categories than the planar network. Since the pre-trained model used for training is based on the planar dataset COCO, the model designed for spherical images has an inherent disadvantage for objects with lower distortion. In addition, some of the categories have a miss rate close to 1.0 among all models, and we conjecture that these categories have too small objects (e.g., cup) or a small number (e.g., clock less than 1%), making it difficult for models to extract features. Overall, our model performs better in terms of the average results. More in-depth, our model performs more robustly across different categories.

The scatter plot in Fig. 4 shows that the mAP values of our model are more evenly distributed across different categories. The planar model Faster-RCNN is more biased, and the accuracy is more differentiated between categories, while the results of SphereNet are leaner towards low values. Fig. 6 shows several examples on 360-Indoor and VOC-360 datasets. The VOC-360 image is projected on 72° polar angles. The planar network Faster-RCNN and the spherical network SphereNet are involved in the comparison, and the proposed method detects more distorted objects. Faster-RCNN shows good performance in regions near the center.

B. Ablation Studies

We performed ablation tests for proposed components on real-world datasets, as shown in Table. II. To be fair, our backbones are all pre-trained with ResNet-50/101 from the COCO dataset without augmentation.

The introduction of the cubemap stream obviously improves the performance of the network. The addition of the Facet-Link module alone improves the mAP by more than two percentage points. The calibration module has a more remarkable improvement on the AP_{75} metric, indicating that the model is

TABLE III

Comparison of different integration strategies. 'CONCAT' denotes concatenation, and '$\oplus$' and '$\otimes$' denote element-wise addition and multiplication. 'Top-x' denotes the mAP value of the category ranked at x-th.

Strategy	mAP (%)	Top-1 (%)	Top-15 (%)	Top-25 (%)
CONCAT	16.66	70.26	36.49	10.41
$\oplus$	15.08	64.51	26.62	2.60
$\otimes$	16.64	64.98	35.39	8.15

able to detect distorted targets with greater confidence. On the contrary, we note that the calibration module has a reduced impact on AP_{50}, probably because the model has a certain recognition of the distorted objects at low confidence like the planar ones.

We perform separately different correlation operations in the feature correction module on the 360-indoor dataset, namely (a) concatenation followed by a convolutional layer for dimensionality reduction, (b) element-wise addition, and (c) element-wise multiplication, as shown in Table. III. We use ResNet-50 as the backbone and adopt the panorama features calibrated by the cube ones as the baseline. We can see that concatenation and element-wise multiplication have comparable effects, while the latter is faster, which is the reason we choose it to obtain better performance for the architecture.

V. Conclusion

In this paper, we propose a cubemap-guided panorama feature rectification method to improve object detection accuracy. The panorama suffers progressively more distortion from the equator to the poles, which the cubemap compensates at the cost of discontinuities. To eliminate discontinuities and form associations between signals, we first link up the semantics between different facets of the cube-domain through the Facet-Link module to capture long-range dependencies as well as the fine-grained features and then pack them for location registration to correct the panorama at the facet level. The experimental results on both real-world and synthetic datasets verify that our method achieves higher accuracy and performs more uniformly and robustly across various categories.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[2] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon et al., "ultralytics/yolov5: v6.1-tensorrt, tensorflow edge tpu and openvino export and inference," Zenodo, Feb, vol. 22, 2022.

[3] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," Pattern Recognition, vol. 130, p. 108796, 2022.

[4] H. Bai, P. Sasikumar, J. Yang, and M. Billinghurst, "A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing," in Proceedings of the CHI conference on human factors in computing systems, 2020, pp. 1–13.

[5] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," Advances in neural information processing systems, vol. 30, 2017.

[6] B. Coors, A. P. Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in Proceedings of the European conference on computer vision, 2018, pp. 518–533.

[7] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical cnns," in Proceedings of the International conference on learning representations, 2018.

[8] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 9181–9189.

[9] Y. Ye, E. Alshina, J. Boyce, "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib," JVET-E1003, Jan. 2017, Geneva, CH.

[10] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2020, pp. 462–471.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[14] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in Proceedings of the IEEE international conference on computer vision, 2019, pp. 6569–6578.

[15] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[17] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in Proceedings of the European conference on computer vision, 2018, pp. 3–19.

[18] F. Kangni and R. Laganiere, "Epipolar geometry for the rectification of cubic panoramas," in the 3rd Canadian conference on computer and robot vision. IEEE, 2006, pp. 70–70.

[19] W. Yang, Y. Qian, J.-K. Kämäräinen, F. Cricri, and L. Fan, "Object detection in equirectangular panorama," in Proceedings of the 24th IEEE international conference on pattern recognition, 2018, pp. 2190–2195.

[20] F. Dai, B. Chen, H. Xu, Y. Ma, X. Li, B. Feng et al., "Unbiased iou for spherical image object detection," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 1, pp. 508–515, Jun. 2022.

[21] P. Zhao, A. You, Y. Zhang, J. Liu, K. Bian, and Y. Tong, "Reprojection r-cnn: A fast and accurate object detector for 360° images," 2019, arXiv:1907.11830.

[22] G. P. de La Garanderie, A. A. Abarghouei, and T. P. Breckon, "Eliminating the blind spot: Adapting 3d object detection and monocular depth estimation to 360 panoramic imagery," in Proceedings of the European conference on computer vision, 2018, pp. 789–807.

[23] Y. Rao, W. Zhao, Z. Zhu, J. Lu, and J. Zhou, "Global filter networks for image classification," Advances in neural information processing systems, vol. 34, pp. 980–993, 2021.

[24] S.-H. Chou, C. Sun, W.-Y. Chang, W.-T. Hsu, M. Sun, and J. Fu, "360-indoor: Towards learning real-world objects in 360deg indoor equirectangular images," in Proceedings of the IEEE winter conference on applications of computer vision, 2020, pp. 845–853.

[25] S. Vicente, J. Carreira, L. Agapito, and J. Batista, "Reconstructing pascal voc," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 41–48.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol. 28, 2015.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan et al., "Microsoft coco: Common objects in context," in Proceedings of the European conference on computer vision, 2014, pp. 740–755.