Viewing Bias Matters in 360° Videos Visual Saliency Prediction

Abstract

360° video has been applied in many areas such as immersive contents, virtual tours, and surveillance systems. Comparing to the field of view prediction on planar videos, the explosive amount of information contained in the omni-directional view on the entire sphere poses additional challenge towards predicting high-salient regions in 360° videos. In this work, we propose a visual saliency prediction model that directly takes 360° video in the equirectangular format. Unlike previous works that often adopted recurrent neural network(RNN) architecture towards the saliency detection task, in this work we utilize 3D convolution to a spatial-temporal encoder and generalize SphereNet kernels to construct a spatial-temporal decoder. We further study the statistical properties of viewing biases present in 360° datasets across various video types, which provides us with insights towards the design of a fusing mechanism that incorporates the predicted saliency map with the viewing bias in an adaptive manner. The proposed model yields state-of-the-arts performance, as evidenced by empirical results over renowned 360° visual saliency datasets such as Salient360!, PVS, and Sport360.

1 Introduction

360° video, a new multimedia type, has become popular due to its immersive experiences (Škola et al. 2020; Radianti et al. 2020; Lee et al. 2020). Compared to conventional planar videos, 360° videos (Neumann, Pintaric, and Rizzo 2000) capture omni-directional field of view in one frame. We can enjoy this immersive experience by drag-and-drop the mouse on social media platforms or changing our head and eye movements with head-mounted display devices. Consumer can even create a 360° video with an off-the-shelf 360° camera, such as Insta360, Samsung Gear360, or Ricoh Theta, and shared them on Facebook or YouTube.

The emerging of 360° videos shows that 360° videos will become a major video format in the near future. However, it's hard for users to explore a whole frame of 360° videos because users cannot observe the content outside the FoV associated with human eyes; users need to change the viewing angles regularly to see the whole 360° environment. To relieve the issue, recent methods of human visual saliency prediction, focusing on modeling human visual attention and identifying users' interest in 360° videos are developed. (Cheng et al. 2018; Zhang et al. 2018; Xu et al. 2018; Zhang et al. 2019).

Nowadays, methods modeling 360° visual saliency(VS) prediction are still limited. Although there are dozens of VS prediction models available for planar videos (Jain et al. 2020; Min and Corso 2019; Chang, Zhu, and Zhu 2021), such models are not suitable to the 360° video which has its own characteristics that must be taken into consideration. First, 360° videos are generally presented in equirectangular projection format which oversamples points in polar regions. Second, in planar videos, a gaussian kernel center bias is usually added into the prediction since users tend to watch the center part of an image (Cornia et al. 2018; Wang et al. 2018). However, since 360° videos capture the whole picture with no dead spots, the human viewing bias might be different from that in planar videos. The viewing bias in 360° videos might be more complex because users can get more content information with multiple viewpoints (Peleg and Ben-Ezra 1999). Besides, users start to watch 360° videos from the same viewing points no matter through VR head mounted devices or Youtube by PC, since the start-viewing point are fixed by 360 cameras. Nevertheless, currently, no visual saliency models possess the initial frame center bias effectively (Zhang et al. 2019) which is an inevitable phenomenon of immersive videos that we found in the statistical data. The data type in 360° VS prediction can be separated into head movement (HM) and head+eye movement (HEM). The former determines the FoV regions seen by users when moving their heads, while the latter predicts users' eye gaze. We focus on HM saliency prediction in this paper since HM could be seen as the first step toward human attention (Xu et al. 2018).

In this paper, we proposed a 3D U-Net VS model with the combination of human viewing biases. Our model applies to equirectangular projection frames directly without any projection transformation. To cope with the oversampled polar regions in equirectangular frames, we introduced the 3D SphereNet specially desinged for 360° data and placed this module into the U-Net decoder. With the observation of the time-decay human viewing bias which is a special phenomenon in panorama videos, we proposed the inital frame center bias fusing methods, and analyzed the viewing biases between various 360° video VS datasets. Finally, with the

aware of viewing bias observations, the Center Bias Fusing Block is proposed to fuse the well-founded viewing biases effectively. Empirical results on Salient360! (David et al. 2018a), PVS (Xu et al. 2018) and Sport360 (Zhang et al. 2018) datasets demonstrate the effectiveness of the proposed method.

The main contributions can be summarized as follows:

- We propose a 3D U-Net VS model and achieve state-of-the-art results on Salient360!, PVS and Sport360 dataset.
- We are the first to expand tailor-made spherical convolution kernel from 2D to 3D and places it in the decoder to deal with spatial-temporal features.
- With the observation of various initial viewing bias between datasets, we propose a learnable time-decay function to fuse the prediction map with different initial viewing bias effectively.
- We design center bias fusion block that improves the result of saliency prediction by considering the center bias statistics between different datasets and video categories.

2 Related Work

2.1 Visual Saliency on Planar Videos

Most visual saliency(VS) prediction on videos have been done by deep convolutional neural networks. Different from VS prediction on images, temporal features are also considered in the VS video task. A majority of recent studies adopted recurrent neural networks to predict sequential fixation maps over successive frames. Wang et al. (2018) applied attentive CNN-LSTM network to extract both static and dynamic saliency features. Wu et al. (2020) extracted temporal information by a correlation-based ConvLSTM (Xingjian et al. 2015) which integrates the correlation information between frames. Droste, Jiao, and Noble (2020) made an unified model that predicts both images and videos with MobileNet and LSTM network.

Recently, 3D convolutional layers are also used in VS prediction tasks on planar videos. Some methods relied on the S3D architecture (Xie et al. 2018) which is a typical action detection backbone. Min and Corso (2019) was the first to introduce the S3D backbone into VS prediction task as the encoder that extracted spatial-temporal information. Jain et al.(2020) also used S3D bockbone as the U-Net encoder and added the auxiliary audio networks to predict audio-visual saliency. Chang, Zhu, and Zhu(2021) proposed feature pyramid network on the S3D encoder features and aggregate multi-scale feature maps to predict visual saliency. All these methods (Min and Corso 2019; Jain et al. 2020; Chang, Zhu, and Zhu 2021) achieved outstanding performance on planar video VS datasets, such as DHF1K (Wang et al. 2018). While ConvLSTM (Xingjian et al. 2015) extracted temporal information only from the hidden state of the the propagation from the succesive previous frames, 3D convolution (Ji et al. 2012) captured the temporal features encoded in multiple adjacent frames.

2.2 Visual Saliency on 360° Videos

Some learning based methods on 360° videos also emerged in the past three years. CubePadding (Cheng et al. 2018) learned the model by weakly supervised learning with optical flow and video frames in the cubemap format. However, CubePadding was not suitable for static videos which do not have much optical flow features. Besides, CubePadding needed extra calculation to transform equirectangular frames to cube faces. Spherical U-Net (Zhang et al. 2018) introduced a spherical convolution, involving the rotation of the crown kernel along the sphere, to tackle with the distortion of 360° videos in equirectangular frames. Spherical U-Net learned the model by teacher-forcing (Williams and Zipser 1989) that the ground-truth of previous frames were fed into the model during training and inferenced the result with previous predictions of the model, causing the performance degraded over time as the prediction becoming less accurate. DHP (Xu et al. 2018) proposed deep reinforcement learning (DRL) approach to predict head movement saliency map in an offline manner. They first transformed a specific subject's FoV regions into rectilinear projection and applied DRL prediction. However, during inference stage, DHP needed to run live fixation points of a specific user and later collected several users' predicted fixation points to generate the saliency maps, which was inefficient. SPN (Zhang et al. 2019) took optical flow and frames in the cubemap format as motion and spatial information and adopted Bi-ConvLSTM to extract temporal features. However, SPN needed extra computational costs on generating optical flow and the cubemap transformation. Although SPN considered human viewing bias by fusing different gaussian prior maps into feature maps by convolution layers, the gaussian prior maps used by SPN were chosen without the support of human viewing analysis in different datasets and video contents. In order to deal with initial viewing bias, SPN fused the average saliency map with optical flow motion features by element-wise product. However, this method ignored the time factor of initial frame viewing bias and the videos with less optical flow features, such as scenery videos.

3 Method

3.1 Network Structure

Spatial-Temporal Encoder. The proposed model architecture is composed of an encoder followed by a decoder with multi-branch skip connections, as illustrated in Fig.1. The encoder is S3D network (Xie et al. 2018) extracting spatial-temporal features through 3D convolution and 3D maxpooling. We use S3D network since it replaces standard 3D convolution with the separable spatial and temporal convolutional blocks which encode the spatial-temporal information efficiently with lower computational costs. Moreover, the pre-trained weight of S3D network on the Kinetics dataset (Kay et al. 2017) which is a large action-recognition dataset is available, making it fast and effective for transfer-learning on the limited visual saliency data. Thus, we use the pre-trained S3D weight as the initial weight of the encoder. The input of the encoder is a sequence of T frames $\{I_{t-T+1}, ..., I_t\}$, where I_t is the equirectangular frame of the video at time t. Then the encoder extracts four scales of spatial-temporal feature maps: F_1, F_2, F_3, F_4 as the later input of the decoder, where F_1, F_2, F_3, F_4 are 1/4x, 1/8x,

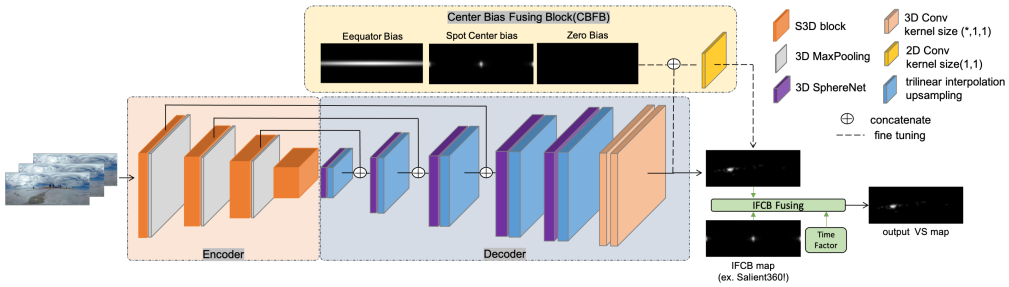


Figure 1: The overview of our model architecture. The encoder consists of S3D block, while the decoder is built with 3DSphereNet layers with trilinear interpolation upsampling layers. The last two layers of the decoder are 1×1 convolutional layers. The Center Bias Fusing Block fuses the three bias maps with the output of the decoder in pixel-wised manner by 1×1 convolution. Finally, the Initial Frame Center Bias(IFCB) module adds the IFCB map of a specific dataset and the time factor dependent on the frame number (see eq.2) to predict the final output VS map.

1/16x, and 1/32x to the original input frames respectively.

360 Kernel Decoder. While the encoder contains standard convolutional kernel, the decoder consists of 3D SphereNet layers which are 3D kernels expanded from SphereNet (Coors, Condurache, and Geiger 2018) used in 360° image classification and object detection. Due to the oversampling around the polar regions in equirectangular projection, we adpot SphereNet kernels which are able to extract repeating features into our decoder module. SphereNet calculates the coordinates of input pixel of convolution by inverse gnomonic projection from the center of kernel. Besides, in order to avoid discontinuities in equirectangular projection, SphereNet automatically wraps the sampling points at the left and right boundaries. However, nowadays there are no spherical kernels deal with both spatial and temporal features, and this is the first work to extend SphereNet from planar kernels to 3D kernels in order to decode spatial-temporal omnidirectional information in video tasks. We sample the input coordinates by gnomonic projection which is adapted in SphereNet among spatial dimension and use trilinear interpolation (Bai and Wang 2010) for sampling among temporal dimension. The inputs of the decoder are F_1,F_2,F_3, and F_4. Except for F_1, the three feature maps: F_2, F_3, and F_4 are passed into the decoder using skip connection and concatenated with the feature maps. F_1 and the concatenated feature maps are decoded by 3D SphereNet layers and are upsampled by trilinear interpolation method. Finally, the decoder outputs a visual saliency prediction map of time t that corresponds to the last frame I_t of the encoder input sequence.

3.2 Initial Frame Center Bias

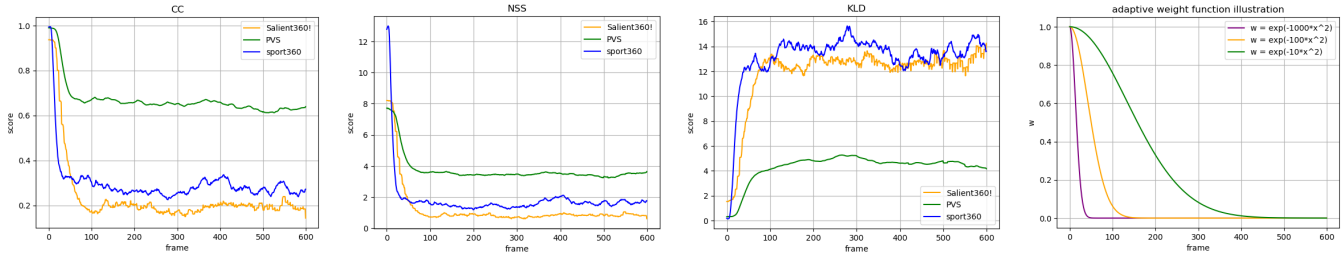
It is a common practice for users to start exploring the 360° videos in the same Field of View (FoV). In other words, users tend to watch the same portion of 360° frames, namely the same longitude and latitude coordinates, at the very beginning of 360 videos. In fact, this initial frame center bias is common in 360° visual saliency dataset, because the start-watching point is determined by the devices, such as 360 cameras. Zhang et al.(2019) considered the initial frame center bias by considering both center bias map and the motion features with residual mechanism. However, this method ignored the vanishing phenomenon of the initial center bias over time.

We designed the initial frame center bias (IFCB) fusing method based on our statistical findings over three datasets: Salient360! (David et al. 2018a), PVS (Xu et al. 2018), Sport360 (Zhang et al. 2018) with the CC, NSS, and KLD metrics; both the dataset descriptions and metric details can be found in Section Experiment. We compute the average of the first visual saliency frames from the training data of the three datasets, which we refer as IFCB map (the top row of Fig.3) and calculated its CC, NSS and KLD scores with ground-truth saliency maps. In Fig.2, the score of IFCB maps from three datasets are shown respectively. Obviously, in all datasets, the CC and NSS scores of IFCB maps are extremely high at the initial frames and gradually degrade as the time increases, which indicates that users spread their view from the same starting point and looked around independently. The low KLD scores at the initial frames also imply that the probability distribution of IFCB map pixels is similar to that of the ground truth at the beginning. Therefore, adding IFCB maps and the time-decay factor into the prediction procedure should be beneficial.

The initial frame center bias of the three datasets are slightly different. In Salient360! dataset, the start-watching points are at longitude $0°$ or the opposite $180°$ (the top row of Fig.3(a)). In PVS and Sport360 dataset, the users all start looking from longitude $0°$ with some latitude offsets (the top row of Fig.3(b)-(c)). Furthermore, in Fig.2, the declining or rising rate of the evaluation scores various between datasets. For example, the CC score of the Sport360 IFCB map decreases drastically from score 1 to 0.4 within 50 frames. On the other hand, the CC score of PVS IFCB map reduces at a slower rate from score 1 to 0.7 within 100 frames. Considering the existing IFCB difference between datasets, we propose an adaptive weighting method to dynamically learn the fusing weights of IFCB map and the decoder prediction map with a time-decaying function. The weighted fusing mechanism can be interpreted as

$$P_t = w_t \times IFCB + (1 - w_t) \times D_t, \quad (1)$$

where D_t represents the output saliency map of the decoder



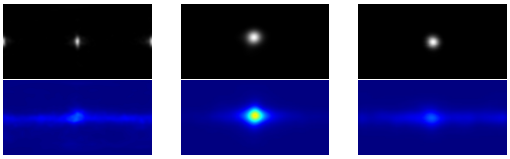
(a) (b) (c) (d)

Figure 2: (a), (b) and (c) are the CC, NSS, KLD score of IFCB map of the three datasets respectively. (d) is an illustration of equation 2. The slope of the curve becomes steeper as alpha increases.

at time t, and P_t is the final prediction saliency map as a linear combination of D_t and IFCB with adaptive weight w_t. Based on the observed time-dependence of the evaluated CC, NSS, KLD scores of IFCB to the ground-truth saliency maps (Fig.2(a)-(b)), we adopt a Gaussian decay function (Fig.2(d)) for the adaptive weight as follows:

$$w_t = \exp(-\alpha(t/C)^2), \tag{2}$$

where C is a constant which we set as 600 in our experiments. Instead of being fixed, α is automatically learned by fine tuning the whole model.



(a) Salient360! (b) PVS (c) Sport360

Figure 3: Top row: IFCB maps. Bottom row: Average saliency maps with frame number over 100.

Dataset \ Video Type	Exploartion	Static	Moving	Rides	Miscellaneous	Number of Video
Salient360!	42.10%	26.31%	15.78%	10.52%	5.26%	19
PVS	14.66%	12.00%	21.33%	28.00%	24.00%	75
Sport360	0.96%	2.88%	76.92%	19.23%	0.00%	104

Table 1: The proportion of video categories in different datasets. *Miscellaneous* refers to videos that do not belong to the four main categories.

3.3 Potential Center Bias

Center Bias Analysis Human attention might have varying viewing bias when watching panoramic videos. In order to have a further observation on human viewing center bias, we analyze the ground-truth saliency maps of each datasets and different video categories and have the following findings. *Finding(1)*: Datasets exist distinct center bias. *Finding(2)*: Fusing different kinds of center bias improves the performance variously in four video categories.

First, we average the saliency maps with frame number over 100 of the three dataset as shown at the bottom row of Fig.3. From Fig.3, we can see that PVS has a strong center bias without doubt. On the contrary, Salient360! has a little bias at the equator and Sport360 has almost no center bias. This indicates that various datasets exist different degrees and distributions of center viewing bias, which is our *Finding(1)*. Second, according to a study in (Almquist et al. 2018) which shows that the Region of Interest (RoIs) that attracts human attention depends on the video content itself, we manually classify the videos of the three datasets into four categories (Table 1):

- *Exploration*: Users tend to explore the entire sphere since there is no particular object or moving direction in scenes, such as landscape.
- *Static Focus*: The salient objects are standstill at the frame center, such as music concert.
- *Moving Focus*: There are eye-catching objects moving over the sphere in the video, such as sport videos.
- *Rides*: Videos are shoot with camera fast moving forward to a specific direction, such as car driving videos.

We average and normalize the ground-truth saliency maps with frame number over 100 of each category, as illustrated in Fig.4. To observe the impact of various center biases on different video categories, we fuse four kinds of center bias maps in Fig.5 which have different coverage on the equator into our Salient360! prediction output with linear combination. Note that the first to third center bias maps in Fig.5(a)-(c) are generated by the equation as follows.

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}\right) \tag{3}$$

The means μ_x and μ_y both equal to 0.5 in the first to third center bias maps Fig.5(a)-(c); the standard deviations are $\sigma_y = 0.02$ and $\sigma_x = 0.5, 0.15, 0.06$, respectively. The fourth center bias map 5(d) is the IFCB map of Salient360!. In Table 2, by fusing the aforementioned four kinds of center bias maps, the CC scores are improved in different degrees depending on the video category. It appears that the *Moving Focus* video type does not benefit from the center bias. This may be because the eye-catching moving objects appear irregularly in various places near the equator. As for the *Exploration* category, the improvement grows as the coverage on the equator becomes larger in center bias map. Since

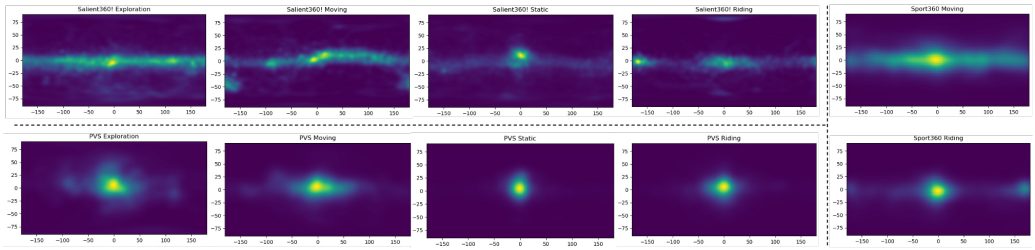
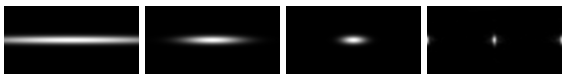


Figure 4: The averaged saliency maps (normalized to $[0, 1]$) with frame number over 100 of four main video catogries over various datasets. Note that we only illustrate salieny maps that are *Rides* and *Moving Focus* types in Sport360 since most of the videos within belong to these two categories, while the *Exploration* and *Static Focus* video types are too few to be of statistical significance.

Exploration type videos lack salient objects, users attention spread along the equator instead of focusing on the same point. The *Static Focus* video type consists of an obvious object in the frame center, so it benefits the most from the center spot bias (Fig.3(c)). Finally, we observe that the *Rides* category videos benefit the most from center bias map that has little dependency on the longitude. Since users tend to watch in the direction of the camera motions, which happen to be at longitude $180°$ in videos that we test in Table 2, the center biases as depicted in 5(a) and 5(d), having values on the $180°$ longitude region, make the more improvement.



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

Figure 5: 5(a)-5(c) are guassian map generated by eq.3 with the same means: $\mu_x = \mu_y = 0.5$ and standard deviation: $(\sigma_x, \sigma_y) = (0.5, 0.02)$, $(\sigma_x, \sigma_y) = (0.15, 0.02)$, $(\sigma_x, \sigma_y) = (0.06, 0.02)$, respectively. 5(d) is the IFCB map of Salient360.

Prior maps / Video type	Fig.5(a)	Fig.5(b)	Fig.5(c)	Fig.5(d)
Exploration	5.52%	3.18%	2.10%	1.51%
Moving Focus	-1.78%	-0.40%	-0.27%	-0.46%
Static Focus	-1.05%	2.47%	4.46%	2.63%
Rides	12.95%	0.69%	0.27%	4.50%

Table 2: The CC score improvement of fusing each center bias maps in Fig.5.

Learned Center Bias Fusing Through the two findings in Section.3.3-Center Bias Analysis, we have a better understanding of the viewing bias in three datasets:

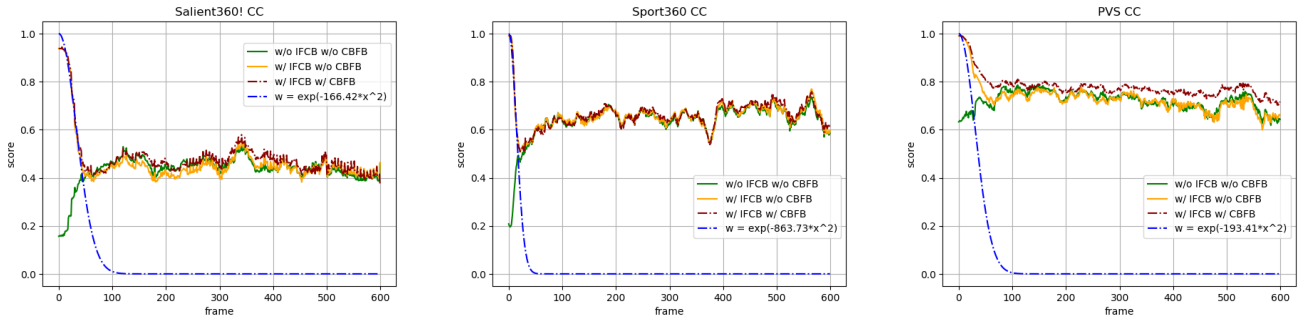
1. Sport360: The video category classification results in Table 1 shows that videos in Sport360 mostly belong to the *Moving Focus* type, which implies that sport360 exhibits nearly no center bias, supported by our *Finding(2)*.
2. PVS: Compared with the other two datasets, the proportion of each video category in PVS is relatively average (Table 1), but the viewing biases have few differences in the four video categories (Fig.5). Obviously, PVS itself exists strong center bias in all video categories, supported by our *Finding(1)*.
3. Salient360!: Different from PVS, Salient360! has a little bias at the equator according to our *Finding(1)*. Besides, there are up to 40% videos belong to *Exploration* type in Salient360! which benefit more from equator center bias. Thus, the equator viewing bias existing in Salient360! can also be partially explained by our *Finding(2)*.

According to the understanding that we conclude above, it is necessary to manage the different extents of center bias among datasets. Through the analysis of composition of the three datasets (Table 1), PVS needs spot center bias more and Salient360! needs equator bias. On the other hand, the viewing data in Sport360! contains nearly no viewing bias. Thus, We introduce Center Bias Fusing Block(CBFB) (Fig. 1). In CBFB, we concatenate equator bias map 5(a), the spot center bias, zero bias map denoting no bias, and the output map of the decoder (Fig.1). Note that we use IFCB map as the spot center bias map since the initial watching regions set by the camera devices are also the center regions of the video. We then pass the concatenation map into an one by one convolution, learning the fusing weight by weighted sum. The CBFB module learn the fusing weights of different viewing biases from the given training data. Finally, the whole model is trained with IFCB and CBFB.

4 Experiment

4.1 Dataset

- Salient360!: The dataset Salient360! (David et al. 2018a) is a benchmark carried out by Salient360! Grand Challenges at ICME'17 and ICME'18 for $360°$ image and video saliency prediction. The benchmark provides 19 equirectangular $360°$ videos each lasting 20 seconds with head movement saliency maps recorded from 57 subjects (David et al. 2018b).
- PVS: PVS dataset (Xu et al. 2018) includes 75 omnidirectional videos each lasting 10 to 80 seconds with head movement saliency maps recorded from 58 subjects. The video contents are diverse, including animation, driving, sports, movies and scenery. The author of PVS splits the data into 60 training videos and 15 testing videos.



(a) (b) (c)

Figure 6: (a), (b) and (c) are the CC score of Salient360!, Sport360 and PVS with and without the IFCB module and CBFB module. The final result of IFCB fusing weights were also shown.

- Sport360: The videos of Sport360 are from (Hu et al. 2017) with the head movement saliency maps collected by (Zhang et al. 2018). Sport360 contains 104 360° sport videos, such as basketball, skateboarding and parkour, with the duration of 20 to 60 seconds viewed by 20+ subjects. Following the settings in (Zhang et al. 2018), we use 80 video sequences for training, and 24 video sequences for testing.

4.2 Implementation Detail

Loss Function Our saliency prediction model is trained by minimizing an integration of several well adopted evaluation metrics. Here we take the combination of *Kullback-Leiber divergence*(KLD), *Pearson's Correlation Coefficient(CC)* and *Normalized Scanpath Saliency (NSS)* metrics as our loss function in the following expression:

$$\begin{aligned} L(P, Q^d, Q^f) = & \lambda_{KL} KL(P, Q^d) - \\ & \lambda_{CC} CC(P, Q^d) - \\ & \lambda_{NSS} NSS(P, Q^f), \end{aligned} \tag{4}$$

where we take $\lambda_{KL} = 2$, $\lambda_{CC} = 0.8$, $\lambda_{NSS} = 0.05$ empirically. The notations are given below:

- P: The predicted saliency map.
- Q^f: The binary fixation ground-truth map that refer to the original fixation locations.
- Q^d: The density distribution ground-truth map that is smoothed by the Gaussian kernel on Q^f (Bylinskii et al. 2018).
- The NSS metric is specially designed for saliency maps (Peters et al. 2005) and is defined as

$$NSS(P, Q^f) = \frac{1}{N} \sum_i \bar{P}_i \times Q_i^f, \tag{5}$$

  where i refers to the i^{th} pixel in Q^f and P respectively. $N = \sum_i Q^f$, and $\bar{P} = \frac{P - \mu(P)}{\sigma(P)}$ where μ and σ are mean and standard deviation of P.
- The CC metric measures the correlation between two distributions as

$$CC(P, Q^d) = \frac{cov(P, Q^d)}{\sigma(P) \times \sigma(Q^d)}, \tag{6}$$

  where $cov(P, Q^d)$ stands for the covariance of P and Q^d, and $\sigma(\cdot)$ denotes the corresponding standard deviation.
- The KLD measures the dissimilarity between predicted saliency and ground-truth distribution which is defined as

$$KL(P, Q^d) = \sum_i Q_i^d \log \left(\epsilon + \frac{Q_i^d}{\epsilon + P_i} \right), \tag{7}$$

  where ϵ is a regularization constant.

Training and Testing Our implementation is on top of PyTorch framework (Paszke et al. 2017). The model is trained in two stages. First, we train the encoder initialized with weights pre-trained on the Kinetics dataset (Kay et al. 2017), and the decoder from scratch until they converge. Then we train the full model in the second stage, including CBFB and IFCB. The Adam optimizer is used with the learning rate 1×10^{-4} at the first stage and 1×10^{-5} at the second stage. The input sequence length is 32 and in equirectangular format without any projection transformation with batch size 4. All frames are resized to 224×384. For Salient360! dataset, we split the data into 15 videos for training and 4 for validation. As for PVS dataset, we randomly split the training data into 50 training videos and 10 validation videos. When training on Sport360 dataset, we choose 10 videos randomly as validation data, and the rest 70 videos as training data. We evaluate our model on the testing videos used in DHP (Xu et al. 2018) and Spherical-Unet (Zhang et al. 2018) of PVS dataset and Sport360 dataset respectively. As for Salient360!, since the ground-truth of the testing data is not available, we evaluate our model on the validatation set.

Evaluation Metric In addition to evaluating our method with KLD, CC and NSS metrics, we also consider *Similarity*(SIM), AUC-Judd and shuffled-AUC metrics (Borji et al. 2013). The details of these metrics can be found in (Bylinskii et al. 2018).

In view of the heavy distortion near the pole regions under equirectangular projections, the Salient360! benchmark corrected the oversampled pole areas by applying a latitudinal sinusoidal factor (David et al. 2018b) to the saliency maps during evaluation. Here we also report the results adjusted by the latitudinal sinusoidal factors with asterisk symbol $(*)$.

4.3 Experimental Result

Ablation Study We preform ablation studies to evaluate the contribution of each component of the proposed network. In Table 3, we compare our modules with standard 3D convolution U-Net. All the components, including 3DSphereNet decoder, IFCB, and CBFB improve the performance in some extent. The 3D SphereNet decoder enhances most of the evaluation metric score except for the KLD/AUC-Judd score in PVS and the CC/NSS score in Sport360. With the combination of 3D SphereNet decoder and IFCB module, the CC score improved about 3.1% on Salient360!, 1.34% on PVS and 1.76% on Sport360. Besides, from Fig.6, the CC scores of the initial frames are raised with the IFCB module. As for CBFB module, the CC scores increase 2.31% on Salient360!, 4.73% on PVS, and 0.36% on Sport360. The different magnitudes of the progress between the three datasets correspond to our first finding that different datasets exhibit distinct degree of center bias (Section.3.3-Center Bias Analysis).

Salient360!					
Method \ Metrics	CC	NSS	KLD	SIM	AUC-J
Standard-3DUnet*	0.3998	1.7924	6.6621	0.3212	0.8749
3DSphereNet*	0.4336	1.9402	6.4355	0.3325	0.8852
3DSphereNet* w/ IFCB	0.4645	2.2537	5.9410	0.3599	0.8891
3DSphereNet* w/ IFCB w/ CBFB	**0.4877**	**2.3502**	**5.7021**	**0.3759**	**0.8927**
PVS					
Method \ Metrics	CC	NSS	KLD	SIM	AUC-J
Standard-3DUnet*	0.6863	3.3031	3.8697	0.5115	0.9280
3DSphereNet*	0.7069	3.3820	3.8869	0.5184	0.9272
3DSphereNet* w/ IFCB	0.7203	3.5330	3.4999	0.5395	0.9295
3DSphereNet* w/ IFCB w/ CBFB	**0.7676**	**3.7498**	**3.2084**	**0.5661**	**0.9325**
Sport360					
Method \ Metrics	CC	NSS	KLD	SIM	AUC-J
Standard-3DUnet*	0.6482	4.4184	5.3503	0.4605	0.9360
3DSphereNet*	0.6449	4.3638	5.0717	0.4654	0.9370
3DSphereNet* w/ IFCB	0.6625	4.5393	**4.8101**	0.4790	0.9375
3DSphereNet* w/ IFCB w/ CBFB	**0.6661**	**4.5860**	4.8529	**0.4793**	**0.9402**

Table 3: The ablation study on the effectiveness of various modules in the proposed model. Standard-3DUnet refers to the U-net model with 3D Convolution, and 3DSphereNet is the 3DUnet with our 3DSphereNet Decoder.

Method \ Metrics	CC	NSS	sAUC
baseline	0.633	3.243	0.519
ViNet	0.633	2.447	0.643
DHP	0.704	3.275	0.700
Spherical U-Net	0.745	3.175	0.700
SPN*	0.767	3.289	0.752
Ours	0.757±0.005	**3.768±0.029**	**0.820±0.004**
Ours*	**0.768±0.005**	3.760±0.031	0.818±0.004

Table 4: The comparison on the testing data of PVS.

Comparison Result We compare our model with state-of-the-art 360° video visual saliency models including ViNet (Jain et al. 2020) dealing with planar video, Spherical U-Net(Zhang et al. 2018), DHP(Xu et al. 2018), and SPN(Zhang et al. 2019). In order to show the effectiveness of IFCB fusing by eq.1, we used IFCB maps directly as our baseline. Our model beats the baseline in all three datasets.

Method \ Metrics	CC	NSS	AUC-J
baseline	0.1761	1.0931	0.2535
ViNet	0.6320	4.3845	0.9244
DHP	0.4445	2.5913	0.8744
Spherical U-Net	0.6246	3.5340	0.8977
SPN*	0.6201	**5.1245**	0.9371
Ours	0.6627±0.003	4.5804±0.039	0.9299±0.002
Ours*	**0.6668±0.003**	4.5674±0.039	**0.9399±0.001**

Table 5: The comparison on the testing data of Sport360.

Method \ Metrics	CC	NSS	KLD	SIM	AUC-J
baseline	0.216	1.130	12.768	0.198	0.402
ViNet	0.400	1.846	6.694	0.314	0.873
DHP	0.175	1.052	15.453	0.2007	0.474
Ours*	**0.483±0.004**	**2.315±0.030**	**5.850±0.113**	**0.369±0.005**	**0.892±0.001**
Ours* (Testing)	0.471	2.087	3.044	0.432	0.817

Table 6: The comparison on the validation data of Salient360!. The last row is our model result on the testing data of the Salient360! benchmark.

Table.4, Table.5 and Table. 6 show the quantitative results of the different methods. Our model outperforms all competitors except for the NSS metric in Table.5. The lower NSS score in Sport360 of our model might be due to SPN also using the optical flow motion as the extra input with cubemap projection format, which enhance the NSS metric that is sensitive to false positive (Bylinskii et al. 2018). We had also reproduced SPN model according to SPN paper description to the best of our knowledge. However, the reproduced results do not meet those reported by (Zhang et al. 2019) in Table 5 (see supplementary materials for more details). Due to the absence of testing ground-truth of Salient360!, we only compare with ViNet and DHP that are reproducible with their open source code on the validation set (Table 6). We also upload the testing result of Salient360! onto the benchmark website, and achieve the best results (Table 6) on CC, NSS, KLD, SIM metrics. [1]

5 Conclusion

In this paper, we address the special phenomenon caused by initial frame viewing bias existing in 360° videos using learnable time-decaying curves, coping with the various time-decay rates among datasets. It is to our observation that datasets need various viewing biases based on the analysis of saliency maps across different datasets, video types and the improvements using multiple center bias maps. Thus, the proposed center bias fusing block learned to find the proper weights of different bias maps of each datasets. We utilize 3D convolution to a spatial-temporal encoder and propose 3D SphereNet ketnerls for the decoder in order to deal with the oversampling of feature maps in polar regions. The proposed method achieve the state-of-the-art results on three publicly available 360° visual saliency datasets, including Salience360!, PVS, and Sport360.

[1] In order to comply with the double blind review policy, we did not cite the Salient360! benchmark website. We will fill up the citation after the reviewing process.

References

Almquist, M.; Almquist, V.; Krishnamoorthi, V.; Carlsson, N.; and Eager, D. 2018. *The Prefetch Aggressiveness Tradeoff in 360° Video Streaming*, 258–269. New York, NY, USA: Association for Computing Machinery. ISBN 9781450351928.

Bai, Y.; and Wang, D. 2010. On the comparison of trilinear, cubic spline, and fuzzy interpolation methods in the high-accuracy measurements. *IEEE Transactions on fuzzy Systems*, 18(5): 1016–1022.

Borji, A.; Tavakoli, H. R.; Sihite, D. N.; and Itti, L. 2013. Analysis of scores, datasets, and models in visual saliency prediction. In *Proceedings of the IEEE international conference on computer vision*, 921–928.

Bylinskii, Z.; Judd, T.; Oliva, A.; Torralba, A.; and Durand, F. 2018. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3): 740–757.

Chang, Q.; Zhu, S.; and Zhu, L. 2021. Temporal-Spatial Feature Pyramid for Video Saliency Detection. *arXiv preprint arXiv:2105.04213*.

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.

Coors, B.; Condurache, A. P.; 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.

Cornia, M.; Baraldi, L.; Serra, G.; and Cucchiara, R. 2018. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10): 5142–5154.

David, E. J.; Gutiérrez, J.; Coutrot, A.; Da Silva, M. P.; and Callet, P. L. 2018a. A dataset of head and eye movements for 360 videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, 432–437.

David, E. J.; Gutiérrez, J.; Coutrot, A.; Da Silva, M. P.; and Callet, P. L. 2018b. A dataset of head and eye movements for 360 videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, 432–437.

Droste, R.; Jiao, J.; and Noble, J. A. 2020. Unified image and video saliency modeling. In *European Conference on Computer Vision*, 419–435. Springer.

Hu, H.-N.; Lin, Y.-C.; Liu, M.-Y.; Cheng, H.-T.; Chang, Y.-J.; and Sun, M. 2017. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1396–1405. IEEE.

Jain, S.; Yarlagadda, P.; Jyoti, S.; Karthik, S.; Subramanian, R.; and Gandhi, V. 2020. ViNet: Pushing the limits of Visual Modality for Audio-Visual Saliency Prediction. *arXiv preprint arXiv:2012.06170*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1): 221–231.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Lee, H.; Jung, T. H.; tom Dieck, M. C.; and Chung, N. 2020. Experiencing immersive virtual reality in museums. *Information & management*, 57(5): 103229.

Min, K.; and Corso, J. J. 2019. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2394–2403.

Neumann, U.; Pintaric, T.; and Rizzo, A. 2000. Immersive panoramic video. In *Proceedings of the eighth ACM international conference on Multimedia*, 493–494.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Peleg, S.; and Ben-Ezra, M. 1999. Stereo panorama with a single camera. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, 395–401. IEEE.

Peters, R. J.; Iyer, A.; Itti, L.; and Koch, C. 2005. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18): 2397–2416.

Radianti, J.; Majchrzak, T. A.; Fromm, J.; and Wohlgenannt, I. 2020. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147: 103778.

Škola, F.; Rizvić, S.; Cozza, M.; Barbieri, L.; Bruno, F.; Skarlatos, D.; and Liarokapis, F. 2020. Virtual Reality with 360-Video Storytelling in Cultural Heritage: Study of Presence, Engagement, and Immersion. *Sensors*, 20(20): 5851.

Wang, W.; Shen, J.; Guo, F.; Cheng, M.-M.; and Borji, A. 2018. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4894–4903.

Williams, R. J.; and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2): 270–280.

Wu, X.; Wu, Z.; Zhang, J.; Ju, L.; and Wang, S. 2020. SalSAC: A video saliency prediction model with shuffled attentions and correlation-based ConvLSTM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12410–12417.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, 305–321.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting.

In *Advances in neural information processing systems*, 802–810.

Xu, M.; Song, Y.; Wang, J.; Qiao, M.; Huo, L.; and Wang, Z. 2018. Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE transactions on pattern analysis and machine intelligence*, 41(11): 2693–2708.

Zhang, Y.; Dai, F.; Ma, Y.; Li, H.; Zhao, Q.; and Zhang, Y. 2019. Saliency Prediction Network for 360° Videos. *IEEE Journal of Selected Topics in Signal Processing*, 14(1): 27–37.

Zhang, Z.; Xu, Y.; Yu, J.; and Gao, S. 2018. Saliency detection in 360 videos. In *Proceedings of the European conference on computer vision (ECCV)*, 488–503.