A Multi-FoV Viewport-based Visual Saliency Model Using Adaptive Weighting Losses for 360° Images

Fang-Yi Chao, Lu Zhang, Wassim Hamidouche, Olivier Déforges

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

A Multi-FoV Viewport-based Visual Saliency Model Using Adaptive Weighting Losses for 360° Images

Fang-Yi Chao, Lu Zhang, Wassim Hamidouche, *Member, IEEE* and Olivier Déforges

The authors are with the Institute of Electronics and Telecommunications of Rennes (IETR) UMR CNRS 6164, National Institute of Applied Sciences, Rennes, France (e-mail: fang-yi.chao@insa-rennes.fr; lu.ge@insa-rennes.fr; wassim.hamidouche@insa-rennes.fr; olivier.deforges@insa-rennes.fr)

***Abstract*—360° media allows observers to explore the scene in all directions. The consequence is that the human visual attention is guided by not only the perceived area in the viewport but also the overall content in 360°. In this paper, we propose a method to estimate the 360° saliency map which extracts salient features from the entire 360° image in each viewport in three different Field of Views (FoVs). Our model is first pretrained with a large-scale 2D image dataset to enable the interpretation of semantic contents, then fine-tuned with a relative small 360° image dataset. A novel weighting loss function attached with stretch weighted maps is introduced to adaptively weight the losses of three evaluation metrics and attenuate the impact of stretched regions in equirectangular projection during training process. Experimental results demonstrate that our model achieves better performance with the integration of three FoVs and its diverse viewport images. Results also show that the adaptive weighting losses and stretch weighted maps effectively enhance the evaluation scores compared to the fixed weighting losses solutions. Comparing to other state of the art models, our method surpasses them on three different datasets and ranks the top using 5 performance evaluation metrics on the Salient360! benchmark set. The code is available at https://github.com/FannyChao/MV-SalGAN360**

***Index Terms*—Human Eye Fixation, Saliency, Omnidirectional Image, Convolutional Neural Network, Deep Learning.**

I. Introduction

VIRTUAL Reality (VR), which is one of the fastest growing multimedia technology in the entertainment industry, attracts many attentions due to its capability of providing users immersive experience in surrounding visual and audio environments. Omnidirectional (or panoramic) images or videos capture all the spatial information in 360° longitude and 180° latitude as a sphere. By wearing Head-Mounted Displays (HMDs), users can freely explore the scene in all directions simply by rotating their heads to different point of views. This interactive property enables users to feel like being in a virtual world. It gives rise to various new challenges at the same time, such as video/image production [1], [2], [3], transmission [4], [5], compression [6], [7], [8], and quality assessments [9], [10], [11]. Those new challenges are dissimilar to the cases in 2D traditional media since users can actively select the content they would like to watch with HMDs, while they are only allowed to passively receive the given content in 2D traditional video. Therefore, the fixation (where users pay more attention) prediction in 360° content becomes essential for user behavior analysis and could benefit 360° VR applications [12], [13]. By leveraging 360° fixation cues and user's orientation sensed by HMD, the performance of 360° streaming system could be significantly enhanced [14]. Saliency prediction in the literature includes two different domains which are fixation prediction and salient object detection. As we focus on fixation prediction in this paper, the term "saliency model" stands for fixation prediction only.

Visual saliency prediction in 360° images can be separated into head movement prediction and head+eye movement prediction [15]. The former predicts the center point in every viewports [16] when users move their heads while watching 360° images. The latter predicts viewer's eye gaze [17]. Although the hypothesis that the center of viewports are observer's eye fixation is followed by [18], [19], authors in Rai *et al.* [20] discovered that the fixation distribution is similar to a doughnut shape distribution which has probability peaks far away from center by 14 degrees. In this paper, we focus on visual fixation prediction based on head+eye movement which outputs a gray scale saliency map presenting the probabilities of every pixel being seen by viewers with no specific intention.

Compared to visual attention models for 360° images, those for 2D traditional images have been well developed in recent years [21], [22], [23]. Seminal methods were proposed based on low-level or high-level semantic feature extraction from handcrafted filters [24], [25], [26] or Deep Convolutional Neural Networks (DCNN) [27], [28], [29], [30], [31] thanks to the establishment of several large scale datasets [32], [33], [34]. Unfortunately, these models are not immediately usable for 360° images because of the severe geometric distortion on the top and bottom areas in equirectangular projection. Furthermore, it is impractical to adjust 2D models simply by training and testing on 360° images because of 1) the lack of a sufficient large 360° image saliency dataset, and 2) the inherent high resolution problem in 360° images whose optimal resolution is at least 3600×1800 pixels recommended by MPEG-4 3DAV group [35] to provide favorable quality. This image resolution excesses the computational limitation of 2D models based on DCNN.

360° image has usually a high resolution as it is constituted of omnidirectional spatial information encompassing a given center. By wearing a HMD, the observer does not look at the entire image at a glance but a small-scale content inside his or her current FoV. Most of existing 360° saliency detection models used the cubic projection to obtain rectilinear images in 90° FoV, but the effectiveness of using 90° FoV has not yet been investigated. Considering Human Peripheral Vision [36] which indicates that 60° FoV is the field that humans have the highest vision acuity, and the most common HMDs (HTC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

Vive and Oculus Rift) on the provide approximated 120° FoV, we first validate the influence of different FoVs in 360° saliency prediction models, then propose a multi-FoV viewport-based model via the integration of salient features extracted from diverse viewport plane image in small (90°×90°), middle(120°×120°), and large (360°×180°) FoV. Our model learns higher semantic content by being pretrained in the 2D image dataset SALICON [33], which is composed of 20000 images. We then fine-tune our model in 360° image dataset with a novel loss function combined with adaptive weighted multiple evaluation metrics to balance the impact of each evaluation factor during the training process.

Different from previous 2D DCNN saliency models [28], [37] trained with single loss function, *e.g.* Binary Cross Entropy (BCE), Normalized Scanpath Saliency (NSS), and Kullback-Leibler Divergence (KLD), [29], [30], [38] established that the combination of three evaluation metrics, *i.e.* KLD, NSS and Linear Correlation Coefficient (CC), provided better results in more evaluation metrics. However, the weightings of these three components were either uniform or manually tuned. Based on a statistical analysis, we propose an adaptive weighting loss function to automatically update the weights during fine-tuning. This method enhances the performance and prevent the time-consuming handcrafted tuning.

Equirectangular projection is the most common method to transfer 360° sphere image into plane image. Considering that it severely oversamples the regions close to the poles, we propose stretch weighted maps attached with adaptive weighting loss function to avoid excessive impact of stretched regions.

We evaluate our model over three public available 360° image datasets, namely Salient360! 2017 [39], Salient360! 2018 [40], Saliency in VR [41]. The results illustrate that our model outperforms the state of the arts on all the tested datasets, and the proposed adaptive weighting loss function enhances the performance by a big margin for some evaluation indexes. The contributions of this paper are summarized as below:

1) Introduce a novel Multi-FoV framework to predict human visual attention: where large FoV feature extraction from the entire 360° image, and middle and small FoV feature extraction from every viewport plane image.
2) Propose an adaptively weighted combination loss function of three evaluation metrics. To the best of our knowledge, this is the first work that dynamically balances the impact of each metric in training process in a saliency prediction architecture.
3) Introduce stretch weighted maps dedicated to viewport-based model with the adaptive weighting loss function to attenuate the geometric distortion in equirectangular format.

The rest of this paper is organized as follows. The current state-of-the-art is discussed in Section II. Our proposed Multi-FoV framework, adaptive weighting losses, and stretch weighted maps are introduced in Section III. The three datasets used for evaluation and other implementation details are described in Section IV. Section V evaluates our proposed model with other state of the arts, and demonstrates ablation studies to validate each part in our architecture. Section VI concludes this paper and presents the future work.

II. Related Work

Previous works on visual saliency prediction for 360° images were inspired by 2D saliency models. They can be categorized into two types: 1) the extensions from 2D models and 2) the tailor-made models for 360° images.

For the models extended from 2D models, their procedures can be comprehended as two parts: preprocessing geometry projection and saliency estimation. Startsev *et al.* [42] interpreted numerous transformations of cubic projection to handle discontinuity problem in saliency map predicted from 2D models for each cube face. Maugey *et al.* [43] projected a 360° image into double cubes, then employed a face detector and a collection of 5 low-level feature extraction models to estimate saliency map. Considering that observers tend to look at more the equator area, Battisti *et al.* [44] projected 360° image into multiple viewports, then both low-level features and high-level features are extracted and averaged to obtain a saliency map refined by an equator-prior weighting map. Lebreton *et al.* [45] proposed a framework called Projected Saliency that combines the adaptive equator-prior map with the saliency map predicted from two existing 2D methods, Graph-Based Visual Saliency (GBVS) [24] and Boolean Map Saliency (BMS) [25], on rectilinear images projected from a 360° image. De Abreu *et al.* presented a postprocessing method motivated by the equator bias tendency in 360° images, called the Fused Saliency Maps (FSM) [19]. It mitigates the undesirable center prior limitation of current 2D saliency models via averaging saliency maps predicted from four horizontal translated 360° images. Zhu *et al.* [46] introduced saliency prediction methods through bottom-up and top-down feature extractions in each projected image for head+eye movement. They applied a stronger equator bias on the head+eye saliency maps to generate head movement saliency map according to their experiments showing that head motions have higher tendency to look at equator area than head+eye motions.

The methods above have a drawback as the 2D models they applied usually have strong center bias located in the center area of projected image. Center bias is a phenomenon describing that humans are used to look at the center of an image to find the most important information [47], [48]. It is feasible for normal 2D images but inappropriate for the images projected from the north pole and south pole of 360° image, since observers usually do not pay attention to them.

Despite of most of 2D model extensions utilizing projection methods to attenuate geometry distortion on equirectangular format, Y. Fang *et al.* [49] and Ling *et al.* [50] used low-level color features extracted from the color contrast between surrounding patches or sub-pixel areas directly in equirectangular 360° images without any projection preprocessing. Thus, these two methods could not avoid geometric distortion inherent in equirectangular format.

The tailor-made models for 360° images are developed exclusively for 360° content and cannot be used for 2D images. Monroy *et al.* [51] built an architecture composed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

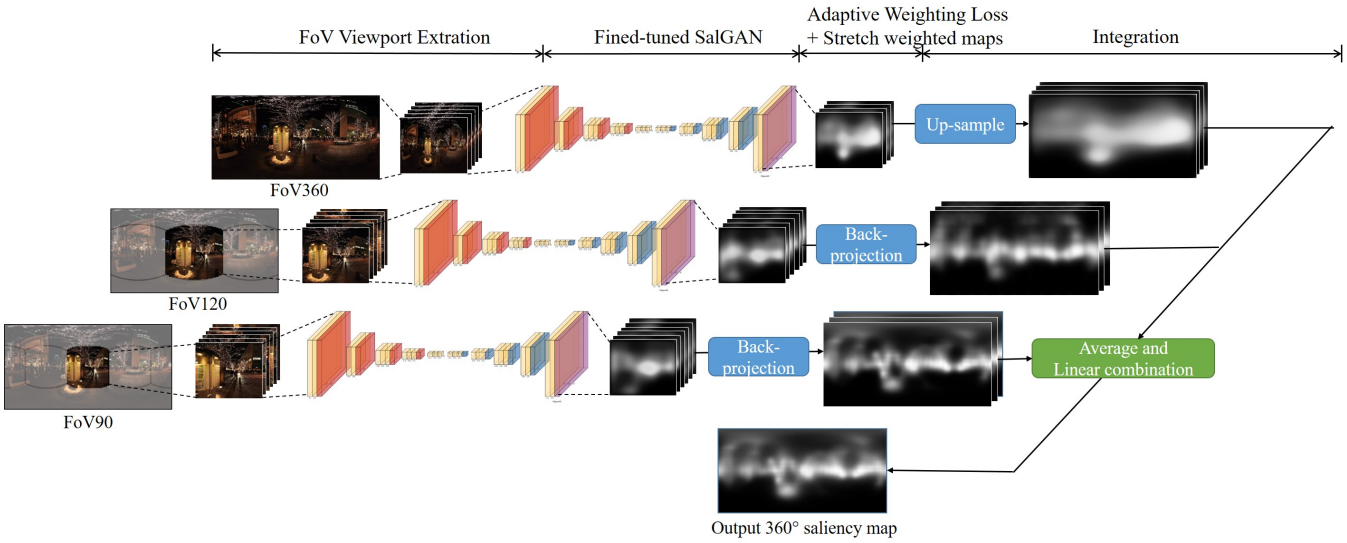

Fig. 1. The overall diagram of our model. The architecture contains three 2D saliency models fine-tuned with our proposed adaptive weighting loss function in three FoVs. It respectively predicts saliency maps of diverse viewport images in each FoV, then the output saliency maps are back-projected to equirectangular format of its corresponding FoV. The final 360° saliency map is linearly integrated from the averaged saliency maps of three FoVs.

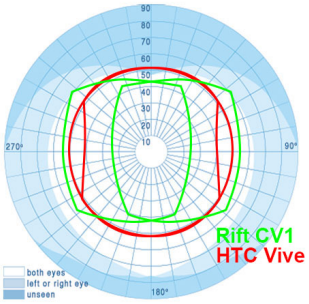

Fig. 2. Region of human FoV and the FoV provided by HMDs. Red circle presents FoV of 2 eyes in HTC Vive, where two straight lines on the left and right side are the edges of FoV of right eye and left eye, respectively. Two green rectangle represent FoV of 2 eyes in Oculus Rift. The largest human FoV reaches 120° vertically and 180° horizontally, but HMDs only provide 120° vertically and horizontally.

of 2 DCNNs. The first network SalNet [52] predicted saliency maps from viewport plane images projected from 360° image. Those saliency maps are then refined by the second network which took account of the corresponding longitude and latitude of each viewport image. As this model computed every viewport saliency map independently, there existed apparent discontinuity in the predicted saliency map back-projected from viewport saliency maps. Chao *et al.* [53] predicted global and local saliency map estimated from multiple cube face images projected from equirectangular image. They fine-tuned a 2D model SalGAN [28] with a loss function combining three evaluation metrics. Cheng *et al.* [54] proposed a weekly supervised method to predict 360° video saliency with Cube Padding (CP) technique which induced no image boundary in DCNN structures by concatenating spatial features in all the six cube faces in the convolution, pooling and convolutional layers of the Long Short-Term Memory (LSTM). However, the proposed video saliency dataset which was used to train their model, is not collected with an eye tracker under the viewing condition of wearing HMD. It was built on the HumanEdit [55] interface, where the annotators see the entire 360° videos in an equirectangular format and use a mouse to direct the FoV. Thus, the dataset data does not correspond to the real user behavior when watching 360° videos. In addition, the model used two image recognition networks (VGG-16 [56] and ResNet-50 [57]) to extract static feature map in each frame and compute the saliency map as the maximum value of feature map. This leads to inaccurate results on other dataset strictly defined by eye fixation map and smoothed with a small view angle. In view of the fact that 360° contents are captured as a sphere then projected to the equirectangular format, Zhang *et al.* [58] proposed a spherical convolution neural network whose kernel was defined on a spherical crown, and the convolution involves the rotation of the kernel along the sphere to extract spherical features without geometry distortion. It down-samples the input image from 1920×3840 to 150×300 in order to save computational memory but leads to abundant important features disappeared.

III. Proposed Model

Human visual saliency is highly related to image scale. For instance, people tend to look at fine details when the image is zoomed in and look at coarse details when the image is zoomed out. When observers wear HMD, she/he does not see the entire 360° image at a glance but only the content inside her/his current viewport. It is similar to the condition that she/he takes a close look to a large image and rotates head to look at other parts of this image. Hence, user visual attention is guided by the salient region not only within the current viewport but also within the overall content in 360° image. According to human visual physiology and the design of the most common HMD, *i.e.* HTC Vive [59] and Oculus Rift [60], on the market, we propose a tailor-made

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

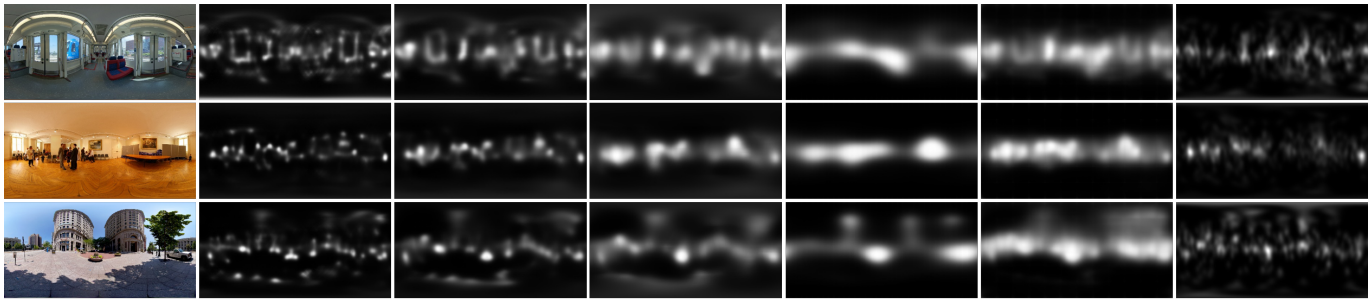

Fig. 3. Saliency maps predicted from FoV60, FoV90, FoV120, FoV360, and our method. Saliency maps from FoV60 falsely detect many fine features while saliency maps from FoV360 overly ignore many of them. Saliency maps from FoV90 and FoV120 are closer to the groundtruth. Our method integrates the saliency maps from FoV90, FoV120, and FoV360 to retain both fine and coarse features.

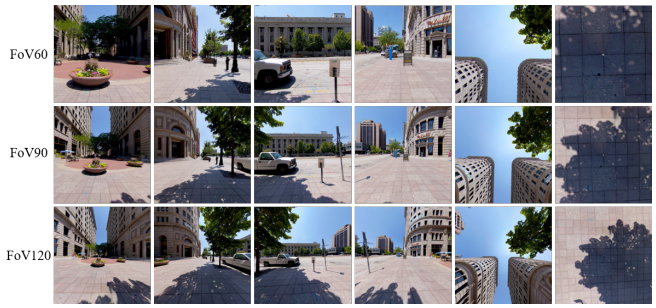

Fig. 4. Rectilinear images of FoV60, FoV90, and FoV120 projected from the bottom left input 360° image in Fig. 3. We can see that images become closer when their FoVs become smaller. According to the designation of HMD, images of FoV120 are closest to what observers see in the HMD.

model taking advantages of three different FoV in low, middle, and high resolutions as input. 360° image is projected into three different FoVs and down-sampled into the same size. Each image is processed by a 2D saliency model, then back-projected to equirectangular format with respect to its FoV size. The estimated saliency maps are linearly integrated by saliency maps yielded by three FoVs.

To alleviate the issue of size limitation of existing 360° image datasets, the 2D saliency model used here is first pretrained in a large scale 2D image saliency dataset SALICON [33], then adjusted in a relatively small 360° image dataset via fine tuning. We propose equirectangular weighted metrics used as loss function to reduce the distortion problem in equirectangular projection caused by upsampling along latitude. Previous 2D saliency models [29], [30] simultaneously took into account several evaluation metrics as the loss function. Instead of manually tuning the weights of each component, we propose an adaptive weighting loss function which updates the weights iteratively during training process. Fig. 1 demonstrates the overall architecture of our model.

A. Multi-FoV and Viewport basis

Fixation prediction in 360° image can be regarded as eye movement in a single viewport and head movement in an entire 360° image. Following eye movement in the current viewport, user's head may rotate to neighboring viewport to look at different contents. Fig. 2[1] visualizes the region of human visual FoV and the FoV provided by HTC Vive and Oculus Rift. It shows that although the largest human visual FoV in horizontal and vertical ranges are respectively about 180° and 120°, the FoV provided by HMD is only about 120° horizontally and vertically. Considering the principle of Human Peripheral Vision [36], which explains the vision occurs outside the fixation point, we define four FoVs as

1) FoV60: It is defined as $60° \times 60°$ in the light of the fact that the highest visual acuity humans have is in the region inside 60° in diameter [36].
2) FoV90: It is defined as $90° \times 90°$ since it is the most commonly used FoV in cubic projection to obtain rectilinear images for 2D extension models.
3) FoV120: It is defined as $120° \times 120°$, which is the FoV that observers perceive instantly in HMD before any movement of eyes and rotation of heads. The scope is due to the designation of HMD.
4) FoV360: It is defined as $360° \times 180°$. Observers are allowed to rotate their head to change viewport. Therefore, all the possible FoV they can see is the entire FoV of 360° images.

To enumerate all the possible points of views that users may look at, we transform a 360° images from equirectangular format to rectilinear images with respect to diverse viewports in each FoV. All these viewport images are down-sampled to the same rectangular size, and served as the inputs to a 2D saliency model for both fine and course features extraction, then the output saliency maps are back-projected to equirectangular format.

Fig. 3 illustrates the saliency maps predicted from 4 FoVs with the 2D saliency SalGAN model. It shows that saliency maps from FoV60 overly detect many fine features, while saliency maps from FoV360 only detect coarse features. Saliency maps from FoV90 and FoV120 are closer to the groundtruth. Fig. 4 illustrates the rectilinear images of FoV60, FoV90, and FoV120 projected from the bottom left input 360° image in Fig. 3. We can see that images of FoV120 contain more information with larger FoV size, and are the closest images observers see in the HMD.

Table I gives the evaluation results of 360° saliency maps predicted from original SalGAN in four FoVs in the test sets of

[1]https://www.reddit.com/r/Vive/comments/4ceskb/fov_comparison/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

TABLE I
EVALUATION OF FOUR FOVS IN TEST SETS OF TWO DATASETS. THE RESULTS IN BOLD AND BLUE COLOR RESPECTIVELY INDICATE THE BEST AND THE SECOND-BEST SCORES ON EACH EVALUATION METRIC. THE SCORES OF KLD ARE THE LOWER THE BETTER, WHILE THE OTHER SCORES ARE THE HIGHER THE BETTER.

Salient360! 2017 [39]						
FoV	KLD↓	CC↑	SIM↑	NSS↑	AUC-J↑	sAUC↑
FoV60	0.627	0.592	0.645	0.481	0.626	0.616
FoV90	0.477	0.648	0.688	0.611	0.665	0.648
FoV120	0.398	0.636	0.699	0.718	0.693	0.689
FoV360	1.236	0.452	0.598	0.810	0.708	**0.752**
FoV60+90+120	0.422	**0.672**	**0.708**	0.666	0.679	0.669
FoV90+120+360	**0.377**	0.654	0.706	**0.850**	**0.719**	0.717
Salient360! 2018 [40]						
FoV60	0.961	0.547	0.609	0.816	0.701	0.629
FoV90	0.737	0.591	0.640	0.874	0.729	0.641
FoV120	0.670	0.586	0.648	0.919	0.741	0.645
FoV360	1.413	0.426	0.555	0.986	0.734	0.588
FoV60+90+120	0.711	**0.617**	0.654	0.927	0.738	**0.649**
FoV90+120+360	**0.650**	0.615	**0.658**	**1.027**	**0.755**	0.645

two datasets. All the test images are projected into rectilinear images with densely cubic projection, which rotates cube in every 10° vertically and horizontally. Then the predicted saliency maps are back-projected into equirectangular format. The six evaluation metrics used in Table I have different natures depending on the definition of saliency and the representation of groundtruth saliency map. Fig. 5 visualizes two saliency groundtruth representations of a 360° image, where fixation map is a binary map recording gaze positions and saliency map is a continuous distribution map presenting the probability of each pixel being seen. We follow the suggestions in [61] to categorise KLD, CC, and Similarity (SIM) into distribution-based metrics as they measure the similarity between predicted saliency map and groundtruth saliency map. NSS, AUC-J, and shuffled AUC (sAUC) are categorised into location-based metrics as they measure how well the predicted saliency map covers the gazes locations in groundtruth fixation map. AUC-Judd is used here as it provides the most accurate approximation to the continuous curve [61], and AUCs is used here to counter the problem of center bias in saliency map [61]. We can see from Table I that FoV90 and FoV120 reach better results while FoV360 reaches the worst on distribution-based metrics (*i.e.,* KLD, CC, SIM) as the distributions of the saliency maps of FoV90 and FoV120 are more close to groundtruth saliency map than that of FoV360. However, FoV360 obtains better results on location-based metrics (*i.e.,* NSS, AUC-J, sAUC) because it covers more fixations in the predicted saliency map. To our surprise, FoV60 generally performs the worst among four FoVs. Its results in distribution-based metrics are worse than that of FoV90 and FoV120 and the results in location-based metrics are worse than that of FoV360. It overly detects excessive fine features in small FoV region and ignores other details on the edges. Thus, it has high false alarm rate and fails to cover groundtruth fixations in many detected salient regions. It does not reach any outstanding result in six metrics. In order to achieve the highest scores in multiple metrics, we propose to integrate multiple FoVs to satisfy different natures of saliency. In Table I, distribution-based metrics (*i.e.,* KLD, CC, NSS) suggest to integrate FoV60, FoV90, and FoV120, while location-based metrics (*i.e.,* NSS, AUC-J, sAUC) suggest to integrate FoV90, FoV120, and FoV360. We report the performance of these two choices in average addition in Table I. We can see that the integration of FoV90+120+360 obtains better results than any single FoV in numerous metrics. Comparing different integrations, the integration of FoV90+120+360 outperforms the integration of FoV60+90+120 in three metrics in the dataset Salient360! 2017 and in four metrics in the dataset Salient360! 2018. It also achieves outstanding results in both distribution-base metrics and location-based metrics. Therefore, we exclude the results of FoV60 and propose a model integrating FoV90, FoV120 and FoV360 together.

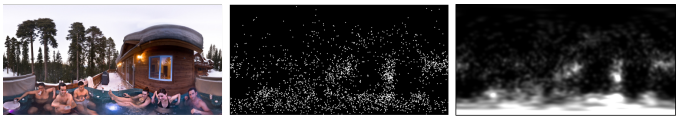

Fig. 5. An example of a 360° image and its groundtruth fixation map and saliency map. Compared to fixation, saliency map oversamples the top and bottom region as it is convoluted in viewport plane and back projected to equirectangular format in the dataset Salient360! 2018 [40].

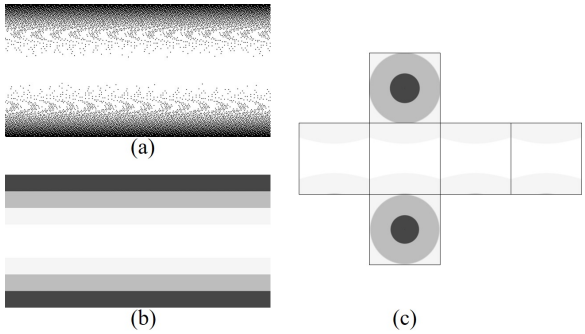

Fig. 6. (a) Helical sphere uniform sample points back-projected to equirectangular format. The number of sample points decreases when it closes to the poles, increases when it closes to equator. (b) Stretch weighted map based on the density of sample points in $N = 8$ regions. The brighter color stands for larger value. (c) Stretch weighted map (b) in cubic format.

We separately fine tune a 2D saliency model with FoV90, FoV120, and FoV360 rectilinear images. For each FoV, 360° equirectangular image is projected to cubic format with its corresponding FoV size as training images. The cube is rotated by every 45° in longitude and latitude to generate more training images in different point of views. Predicted saliency maps from each fine tuned model are back-projected and averaged to equirectangular format, then these three saliency maps are linearly integrated to produce the final saliency map.

B. Stretch Weighted Maps

As the equirectangular image is stretched in the north and south poles of a sphere, we propose stretch weighted maps applied in the loss function in fine tuning process to avoid excessive impact of stretched regions. Fig. 6 demonstrates the sample points, based on helical sphere uniform sampling [62], back-projected to equirectangular image and its corresponding stretch weighted maps in equirectangular and cubic formats, respectively. The brighter color stands for larger value. The stretch weighted map is divided into N regions, and the density

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

of sample points in each region n is computed as weighted value. It is defined as

$$W_i = \frac{\sum_{i \in n} Samplenum\_n}{Total\_Samplenum\_n}, \quad n = [1, ..., N] \tag{1}$$

$$W_i = W_i / max(W) \tag{2}$$

where i denotes each pixel in the map, N denotes the number of regions, n denotes the region that pixel i belongs to, $Sample\_num$ and $Total\_Sample\_num$ denote the number of sample points in a region and in the entire map, respectively. Then the entire map W is normalized with its maximum value. Therefore, the region closer to the poles has lower weight, and the region closer to the equator has larger weight.

Previous works [58], [63], [64] proposed spherical representations for DCNN models, where Zhang *et al.* [58] designed a novel bowl-shape convolutional filter (*i.e.,* kernel) to extract spherical features in sphere images, Coors *et al.* [63] designed convolutional kernel which samples corresponding locations of kernel elements in equirectangular image based on longitude and latitude, and Eder *et al.* [64] proposed a spherical representation based on the icosahedral Snyder equal-area (ISEA) projection and used kernel proposed in [63]. The advantage of these methods is that they can extract spherical features with less geometric distortion. However, they are not suitable in our viewport-based framework which simulates the true viewing scenario in HMD where observers only see viewport plane images. Unlike them, our stretch weighted maps have two advantages: 1) they can be directly used on any pretrained 2D saliency models. No need to train with new kernels like [58]. 2) they can retain high resolution in viewport images. For example, the spherical representation methods mentioned above encode the entire sphere images in dimension $H \times W$. Our method can keep the dimension $H \times W$ in every viewport image, and the back-projected equirectangular map can be in dimension $2H \times 4W$. In particular, the spherical representation of bowl-shape kernel [58] downsamples equirectangular images from 1920×3840 to 150×300 in saliency detection model. Table V compares its performance in dataset Salient360! 2017 [39]. We can see that it does not outperform other state of the arts. In this end, our stretch weighted maps can be seen as a simple method to reduce geometric distortion in rectilinear images trained on 2D models and retain high resolution for 360° images.

C. Adaptive Weighting Loss Function

Plenty of evaluation metrics are available to score the predicted saliency map according to the definition of the saliency and the representation of the groundtruth map [61]. The groundtruth of each 360° image includes a binary fixation map recording user's gaze positions and a continuous saliency map presenting the probability distribution post-processed by convoluting each fixation location via a Gaussian filter with its standard deviation equal to human visual angle.

Fig. 5 shows a 360° image and its groundtruth fixation map and saliency map. We can see that saliency map suffers from serious geometric distortion in the bottom region as convolution is applied on viewport plane, then back-projected to equirectangular format in the dataset Salient360! 2018 [40]. Thus, the stretch weighted maps introduced in Section III-B should be used with saliency maps in the loss function for reducing the impact of geometric distortion. The KLD measures the dissimilarity under the loss of information between predicted saliency distribution and groundtruth distribution. It is defined as

$$L_{KLD}(P, Q^D) = \sum_i Q_i^D \log\left(\frac{Q_i^D}{P_i + \epsilon} + \epsilon\right) \tag{3}$$

where P and Q^D indicate the density distributions of the predicted saliency map and the groundtruth, respectively. i represents the ith pixel and ϵ is a regularization constant. The lower value of KLD means the higher similarity of two distributions. As saliency map which oversamples the top and bottom region as shown in Fig. 5, we introduce stretch weighted map to KLD as

$$L'_{KLD}(P, Q^D) = \sum_i W_i Q_i^D \log\left(\frac{W_i Q_i^D}{W_i P_i + \epsilon} + \epsilon\right) \tag{4}$$

The CC symmetrically calculates the linear relationship between two distributions. It penalizes false positives and false negatives equally. It is defined as

$$L_{CC}(P, Q^D) = \frac{\sigma(P, Q^D)}{\sigma(P) \cdot \sigma(Q^D)} \tag{5}$$

where $\sigma(P, Q^D)$ is the covariance of P and Q^D, $\sigma(P)$ and $\sigma(Q^D)$ are the standard deviations of P and Q^D, respectively. The value of CC ranges from -1 to $+1$, where $+1$ indicates a perfect correlation, and -1 indicates a perfect correlation in opposite direction, and 0 indicates no correlation. It can be introduced with our weighted map as

$$L'_{CC}(P, Q^D) = \frac{\sigma(WP, WQ^D)}{\sigma(WP) \cdot \sigma(WQ^D)} \tag{6}$$

The NSS measures the correspondence between predicted saliency map and groundtruth binary fixation map via computing the average of normalized predicted saliency map at fixation locations. NSS is defined as

$$L_{NSS}(P, Q^B) = \frac{1}{N} \sum_i \frac{P_i - \mu(P)}{\sigma(P)} \cdot Q_i^B \tag{7}$$

where Q^B indicates the groundtruth binary fixation map, i indicates the i^{th} pixel, and N is the total number of fixated points. NSS of value 0 represents chance and positive value represents the correspondence above chance and negative value represents anti-correspondence. Note that unlike KLD and CC whose groundtruth are saliency maps, NSS uses the groundtruth of fixation maps which directly records the locations of gaze points in longitude and latitude in equirectangular format without any oversampling. There is no need to apply our stretch weighted maps in NSS.

For accomplishing the best performance in every evaluation element, a combination of three metrics (KLD, CC and NSS) is used [29], [30], [38] to simultaneously take account of different factors. However, the weights of these three components were either fixed to equal or manually tuned by their scales.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

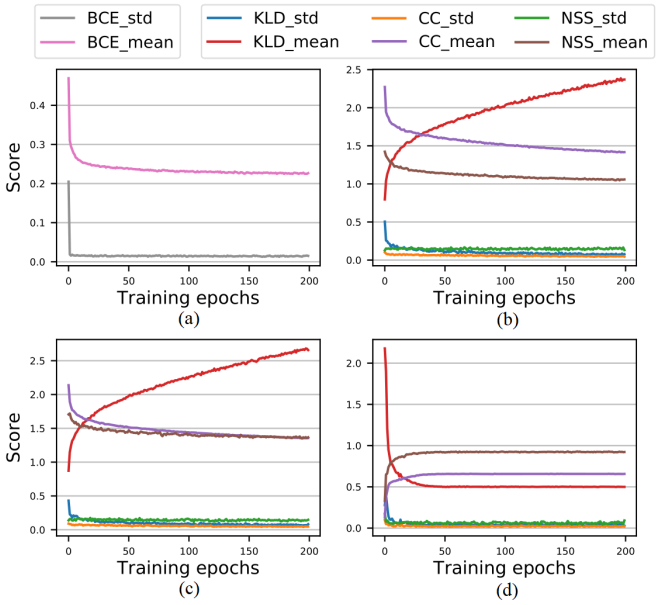

Fig. 7. Standard deviation and mean of each component in loss function in training epochs, where (a) is SalGAN trained with BCE, and (b), (c), (d) are pretrained SalGAN fine tuned with combination loss function with fixed weights $KLD - CC - NSS$, $10KLD - 2CC - NSS$, and our adaptive weights on training set of Salient360! 2017 [39] in FoV90, respectively.

Considering that to manually tune the weights is too time-consuming, and actually the scale of each component keeps changing during training process, we propose an adaptive weighting method to dynamically balance the influence of each component in DCNN models. Our loss function can be interpreted as

$$L = \frac{1}{\sigma_{KLD'}} L'_{KLD}(P, Q^D) - \frac{1}{\sigma_{CC'}} L'_{CC}(P, Q^D) - \frac{1}{\sigma_{NSS}} L_{NSS}(P, Q^B) \quad (8)$$

where $\sigma_{KLD'}$, $\sigma_{CC'}$, and σ_{NSS} are the standard deviations of L'_{KLD}, L'_{CC}, and L_{NSS} calculated with all training images in all iterations in current training epoch and updated in the next training epoch. Note that the standard deviations here are not calculated and applied to Equation (8) in current iteration as their values could be very small in current mini batch and cause extremely large loss in Equation (8) that results in excessive gradient decent. Hence, our method calculated standard deviations in all training images to attain general scales of KLD, CC, and NSS in current epoch and apply to Equation (8) in the next epoch. They can be regarded as the weightings which constrain each component in Equation (8) in the same scale and have equal impact to this loss function. The weighted KLD and weighted CC are used here to decrease the impact of stretched regions and increase the impact of unstretched regions. We give positive weighting to L'_{KLD} and negative weighting to L'_{CC} and L_{NSS} since L'_{KLD} should be minimized while L'_{CC} and L_{NSS} should be maximized, so that the overall loss function L is minimized. As we measure the standard deviations of weighted KLD, weighted CC and NSS in each epoch, along with more training epochs are updated, the value of standard deviation σ decreases and the value of loss function increases. An adaptive learning rate is used here to prevent excessive gradient decent. The decay rate of learning rate is defined as $(1 - epoch/max\_epoch)^{\lambda}$, which becomes smaller and smaller during training.

As parameters in a DCNN model can be learned with back-propagation according to the loss between output and groundtruth, the mean and the standard deviation of loss decrease in every training epoch and the predicted saliency map gradually approaches to the groundtruth during training process. For example, Fig. 7 (a) visualizes the standard deviation and the mean of BCE in every training epoch when SalGAN [28] is trained with BCE on dataset SALICON [33] as suggested in its paper [28]. It shows that the mean and standard deviation of BCE keep decreasing with more and more training epochs, and indicates that the predicted saliency maps are getting closer to the groundtruth. Fig. 7 (b), (c), (d) illustrate the mean and standard deviation of three components in combination loss function with equal weights (*i.e.,* $KLD - CC - NSS$) used in [30], manually tuned weights (*i.e.,* $10KLD - 2CC - NSS$) used in [29], [38], and our adaptive weights (*i.e.,* Equation (8)) of SalGAN model fine tuned on dataset Salient360! 2017 [39] in FoV90. Note that the scores of KLD are the lower the better, while the scores of CC and NSS are the higher the better. We can see that in Fig. 7 (b) and (c), the mean of KLD does not decrease and the mean of CC and NSS do not increase in training epochs. It implies that these three components are not optimized as linear sum of losses could directly converge to zero during training process. In contrast, Fig. 7 (d) shows that the mean of KLD decreases and the mean of CC and NSS increase as what we expect in training epochs. It explains that our adaptive weighting loss function is able to successfully optimize three evaluation metrics and simultaneously achieve the best converged results in these three components.

D. Integration of three FoVs

With the stretch weighted maps and adaptive weighting loss function in fine tuning, 2D saliency models can be adopted to 360° images. As shown in Table I, three FoVs have highest scores in different evaluation metrics. To achieve good performance in all the metrics, we use the linear additive formula to combine these maps. After the model separately fine tuned by three FoVs generates three saliency maps, we linearly integrate them as

$$S = \alpha S_{FoV90} + \beta S_{FoV120} + (1 - \alpha - \beta) S_{FoV360} \quad (9)$$

where S denotes the final saliency map, S_{FoV90}, S_{FoV120}, and S_{FoV360} refer to saliency maps predicted from FoV90, FoV120 and FoV360, respectively.

IV. Experimental Setup

Our model is implemented on top of SalGAN [28] framework due to its powerful yet simple architecture. It detects the saliency map with Generative Adversarial Network (GAN) including a generator to predict saliency map and a discriminator to distinguish the authenticity of predicted map. In this section, we describe the experimental settings, datasets and metrics used for evaluation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

A. Datasets

To ensure a comprehensive comparison, we use 3 datasets with different image contents, different acquisition equipments and saliency maps generated in different ways to evaluate our method. We list the descriptions of these datasets in the following:

- **Salient360! 2017** [39]: This dataset released 60 omnidirectional images which contains 20 images for head movement and 40 images for head+eye movement to the public for free-use, and 25 omnidirectional images containing both head, and head+eye movement for evaluating in ICME2017 challenge. In order to equally compare our model with others, we follow the rules of this challenge to train our model with free-use 40 images for head+eye movement and evaluate with 25 images used in the challenge. All the images are in equirectangular format with resolutions ranged from 5376×2688 pixels to 18332×9166 pixels. Fixation locations and head positions of each image are collected from at least 40 observers seating in a rolling chair, wearing HMD Oculus-DK2 and watching each image for 25 seconds. The starting position is set in the center of images at the beginning of each visualization. A small eye-tracking camera is embedded in HMD to record fixation of dominant eye at 60 Hz. A Gaussian of 3.34° visual angle is applied to blur all the fixation points within the viewport plane, then back-projected to the final equirectangular saliency map.
- **Salient360! 2018** [40]: It was built similar to Salient360! 2017 but the authors improved some aspects of the processing of raw data and generation of saliency maps (e.g. using information for the two eyes, and some more). That is why the provided saliency maps are very different from the Salient360! 2017 dataset. There are 101 equirectangular omnidirectional images and their saliency map and fixation maps in this dataset. The groundtruth of 85 images was released to public for the training and the validation purpose, while 26 images was kept secretly for the test and the benchmark [65]. We give our method to the authors to get its performance on the test images and compare it with other state of the art methods with known performance (on the benchmark website) but without paper to be referred to.
- **Saliency in VR** [41]: 22 panoramas including indoor and outdoor scenes are used to record 122 users' eye fixation under three different viewing conditions: viewed with HMD in a standing or seating position in a non-swivel chair, and seated in front of a desktop monitor. We only consider the standing condition in this paper as users are more willing to move and rotate their heads in the standing position. All the panoramas were viewed in 30 seconds began in the different starting points. Fixations were recorded with a pupil-labs1 stereoscopic eye tracker installed in Oculus DK2 HMD at 120 Hz. Fixation maps were convolved by a Gaussian with standard deviation of 1° visual angle to yield continuous saliency maps. Panoramas viewed with HMD at the same start point standing and seating are used in our comparison.

B. Evaluation Metrics

We consider 4 evaluation metrics which are KLD, CC, NSS, and AUC-Judd [61] here. KLD, CC, NSS are the same as the demonstration in Section III-C. The Area Under Curve (AUC) computes the area under the curve of true positive rate versus false positive rate for various level-set thresholds. The prediction ability of a saliency map is evaluated by how many groundtruth fixations it captures in the successive thresholds. In this paper, we use the AUC-Judd, where the saliency map serves as a binary classifier of fixations at various thresholds, and the threshold values are the saliency values at fixation locations. The true positives are the summation of saliency values at fixated pixels above threshold, and the true positive rate is the proportion of true positives values to the total number of fixations. The false positives are the summation of saliency values at unfixated pixels above threshold, and the false positive rate is the proportion of false positives values to the total number of saliency map pixels at a given threshold.

Note that it is incorrect to compare two saliency maps in equirectangular format since it oversamples the points close to the north pole and south pole. Therefore, we abide by the comparison method used in the Challenge Salient360! 2017 [39] and Salient360! 2018 [40] which only compares predicted saliency map and groundtruth map with the sampled points uniformly distributing on a sphere.

C. Training and Testing

SALICON dataset is used to pretrain our method in Sal-GAN framework. The hyper parameters follow the suggestions from [28]. Our model is then fine-tuned on the dataset Salient360! 2017 via transfer learning. 30 images are used for training, 10 images for validation and 25 images for evaluation. In training process, the adaptive learning rate is set $\lambda = 0.9$ by experiment. In validation and test processes, rectilinear images of each FoV are projected in every 10° along longitude and latitude to enumerate all possible viewports that observer may see. Then predicted viewport saliency maps are back-projected and averaged into an equirectangular map. Three equirectangular saliency maps predicted from FoV90 and FoV120 are linearly integrated with the saliency map estimated from the FoV360 into a final equirectangular map. Integration parameters here are $\alpha = 0.2$, $\beta = 0.5$, $(1-\alpha-\beta) = 0.3$ from the best result achieved in validation images. For stretch weighted map, we sample $4 \times \pi \times 4000$ points in 192×384 map, and set $N = 8$ regions.

V. Experimental Evaluation

Each component in our architecture is analyzed to validate its contribution with the dataset Salient360! 2017. Therefore, all the evaluation results in the experiments are tested on its 25 evaluation images. The quantitative and qualitative comparisons with other state-of-the-art models are also performed in this section.

A. Stretch Weighted Maps and Adaptive Weighting Loss

We evaluate the results of the models with/without fine-tuning with adaptive weighting loss function, BCE used in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

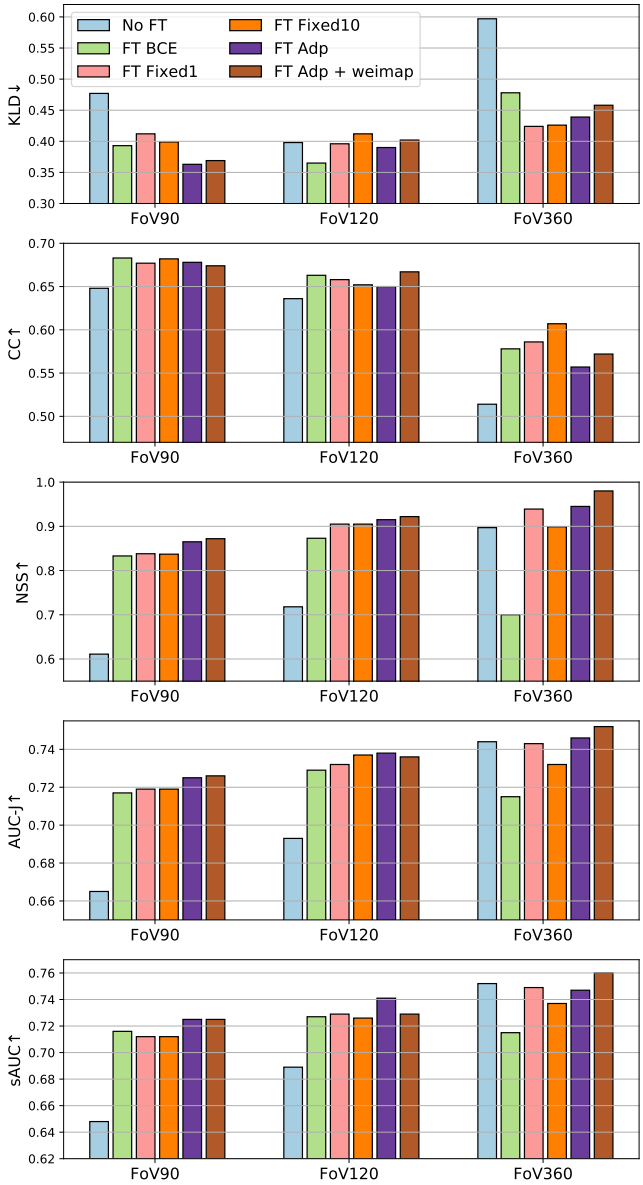

Fig. 8. Evaluation of the models with/without fine-tuning with BCE, two fixed weighting loss functions, and adaptive weighting loss function evaluated on the test set of Salient360! 2017 [39]. no FT stands for no fine tuning, FT BCE stands for fine tuning with BCE, FT Fixed1 stands for fine tuning with $KLD - CC - NSS$, FT Fixed10 stands for fine tuning with $10KLD - 2CC - NSS$, FT Adp stands for fine tuning with adaptive weighting loss, and FT Adp + weimap stands for fine tuning with adaptive weighting loss and stretch weighted maps. Lower KLD value indicates a better performance, and a higher score of other metrics means a better performance.

SalGAN, and two fixed weighting loss functions, which are $KLD - CC - NSS$ used in [30] and $10KLD - 2CC - NSS$ used in [29], [38].

In Fig. 8, we can see that fine-tuning, no matter with adaptive weighting, or fixed weighting, enhances the performance of the four metrics. Comparing the BCE and the combination loss functions (adaptive weighting and the fixed weighting), BCE only has good performance in KLD and CC, and the combination loss functions are better in NSS and AUC-Judd. To our surprise, fine tune with BCE has worse performance than without fine tune in NSS and AUC-Judd in FoV360. Comparing adaptive weighting and the fixed weighting loss functions, two fixed weighting loss have similar performance in four evaluation metrics, while adaptive weighting has better performance in NSS and AUC-Judd. For stretch weighted maps, it improves the performance of adaptive weighting loss in CC and NSS, but slightly worsens the performance regarding KLD. In General, adaptive weighting loss function with stretch weighted maps attains relatively superior performance compared with others in four evaluation metrics.

Comparing three FoVs, fine tuned FoV90 with stretch weighted maps is outperforming in KLD and CC, but underperforming in NSS and AUC-Judd, while fine tuned FoV360 with stretch weighted maps is outperforming in NSS and AUC-Judd, but underperforming in KLD and CC. Fine tuned FoV120 with stretch weighted maps performs in the middle of the other two.

TABLE II
RESULTS OF DENSELY CUBIC PROJECTION IN THREE ROTATION ANGLES EVALUATED ON THE TEST SET OF SALIENT360! 2017 [39]. THE RESULTS IN BOLD INDICATE THE BEST SCORES.

Rotation Angle	KLD↓	CC↑	NSS↑	AUC-J↑	Computation Time (sec)
90°	0.653	0.666	0.839	0.706	0.02
30°	0.467	0.672	0.869	0.722	0.21
10°	**0.369**	**0.674**	**0.872**	**0.726**	1.86

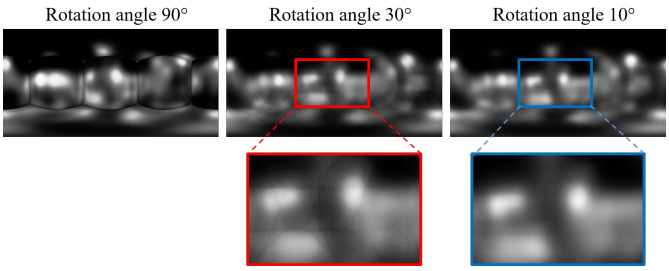

Fig. 9. An example of saliency maps generated from densely cubic projection in rotation angle 90°, 30°, and 10°, respectively. We can see that there are obvious discontinuous borderlines in saliency map generated from 90°, slight discontinuity in 30°, and almost no discontinuity in 10°.

B. Densely Cubic Projection

As mentioned in Section II, numerous methods proposed different projection strategies to reduce discontinuity problem between each viewport. Fig. 12 and Fig. 13 illustrate the results of three state-of-the-art methods [19], [51], [42]. Unfortunately, we can still observe some unfavorable borderlines of viewports in their output saliency maps. We therefore utilize densely cubic projection which samples viewports in every 10° in longitude and latitude, then back-projects viewport saliency maps into equirectangular format and averages equirectangular maps to output a final saliency map. In this way, the unfavorable borderlines can be removed by averaging viewports in different angles. Fig. 9 presents an example of saliency maps generated from rotation angle 90°, 30°, and 10°, respectively. We can see from this figure that rotation angle 90° causes obvious discontinuous borderlines between cubic faces, rotation angle 30° obtains smoother results but still contains slight discontinuity when we take a closer look. Rotation angle 10°, which is used in our method, attains the smoothest result compared to that of 90° and 30°. As

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

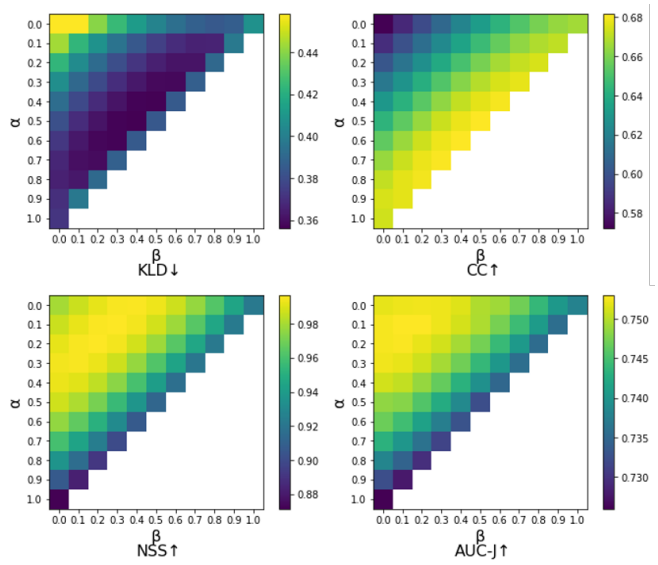

Fig. 10. The results of all the (α, β) combinations in equation (9) in KLD, CC, NSS, and AUC-J evaluated on the test set of Salient360! 2017 [39]. Saliency maps from FoV90, FoV120, and FoV360 here are generated from the model fine tuned with adaptive loss function and stretch weighted maps. No one combination of (α, β) is able to reach the highest scores in all the four metrics.

the smaller rotation angle is, the smoother output saliency maps are. However, more viewport images should be generated increasing the computation time per output saliency map. For example, rotation angle 90° generates 6 viewport images, rotation angle 30° generates 3×3×6 = 54 viewport images, and rotation angle 10° generates $9 \times 9 \times 6 = 486$ viewport images. Table II lists the performance and computation time per output saliency map in rotation angle 90°, 30°, and 10°, respectively. The used computer is equipped with a Intel i9-7900X CPU processor and a 64GB RAM. Even though rotation angle 10° achieves the best scores in four evaluation metrics, it takes the longest time and its performance does not improved much from that of rotation angle 30°, except for KLD which has 27% improvement. Since our model is proposed for 360° image, not video containing numerous frames, computation time is not our primary consideration and 1.86 sec per saliency map is acceptable for us. Hence, we choose rotation angle 10° on account of its good performance.

C. Integration of Multiple FoVs

As three FoVs have their highest and lowest scores in different evaluation metrics, a linear addition of three FoVs is proposed to reach the best performance in all the metrics. In Fig. 10, we assess all the possible combinations of integration parameters (α, β) in Equation (9) in KLD, CC, NSS, and AUC-Judd. In KLD, the three FoVs integration obtains lower scores compared to two FoVs integration and one single FoV. Although FoV360 has the highest score, it decreases the score when it is slightly added to FoV90 + FoV120. The smallest KLD score happens in 0.5FoV90 + 0.4FoV120 + 0.1FoV360. In CC, FoV90 + FoV120 provides the best performance compared to three FoVs integration and one single FoV. The highest CC score is obtained by 0.6FoV90 + 0.4FoV120. In NSS, high scores occur when FoV360 has large parameter

TABLE III
THE EVALUATION SCORES OF EACH FOV AND THE BEST COMBINATION OF (α, β) IN EQUATION (9) OF ANY TWO FOVS INTEGRATION, AND THREE FOVS INTEGRATION ON THE TEST SET OF SALIENT360! 2017 [39]. $\gamma = (1 - \alpha - \beta)$. R REFERS TO THE TOTAL RANKING SUMMED WITH EACH SCORE RANKING LISTED IN PARENTHESES IN GRAY COLOR. THE RESULTS IN BOLD INDICATE THE BEST COMBINATION WHICH HAS THE HIGHEST RANKING.

(α, β, γ)	KLD↓	CC↑	NSS↑	AUC-J↑	R
(1, 0, 0)	0.369 (2)	0.674 (2)	0.872 (7)	0.726 (7)	18
(0, 1, 0)	0.408 (6)	0.649 (4)	0.916 (6)	0.738 (5)	21
(0, 0, 1)	0.458 (7)	0.572 (7)	0.980 (3)	**0.752** (1)	18
(0.4, 0.6, 0)	0.385 (4)	**0.680** (1)	0.917 (5)	0.735 (6)	16
(0.5, 0, 0.5)	0.381 (3)	0.643 (5)	0.988 (2)	0.749 (3)	13
(0, 0.5, 0.5)	0.402 (5)	0.633 (6)	**0.993** (1)	0.750 (2)	14
(0.2, 0.5, 0.3)	**0.363** (1)	0.662 (3)	0.978 (4)	0.747 (4)	12

and integrated with relatively small FoV90 and small FoV120. 0.1FoV90 + 0.3FoV120 + 0.6FoV360 generates the highest NSS score. In AUC-Judd, it is similar to NSS that a large parameter of FoV360 combined with small FoV90 and small FoV120 acquires high scores. The best score lies in 0.1FoV90 + 0.1FoV120 + 0.8FoV360. We can see in Fig. 10 that there is no one (α, β) able to reach the highest scores in all the four metrics.

In order to achieve good results in all the evaluation metrics and prevent biases on some factors, we rank the scores and choose the combination which makes all the four evaluation scores above their median values. Table III reports the performance of each single FoV and the best combinations of any two FoVs integration and three FoVs integration by selecting the highest ranking sum of four evaluation metrics. $\gamma = 1-\alpha-\beta$ and R represents the total ranking summed by the ranks of each score listed in parentheses in gray color. In the table, (α, β, γ) = (1, 0, 0), (0, 1, 0), and (0, 0, 1) are the results of each single FoV, where (1, 0, 0) stands for FoV90, (0, 1, 0) stands for FoV120, and (0, 0, 1) stands for FoV360. They can be regarded as the upper bound performance of each FoV in our model. Besides, $(\alpha, \beta, \gamma) = (0.4, 0.6, 0)$ represents the best result among all the combinations of FoV90 + FoV120, (0, 0.5, 0.5) represents the best result among all the combinations of FoV120 + FoV360, and $(0.5, 0, 0.5)$ represents the best result among all the combinations of FoV90 + FoV360. They can also be regarded as the upper bound performance of any two FoVs integration in our model. The best result among all the combinations of three FoVs integration is also listed here as (0.2, 0.5, 0.3). We rank all the scores in each metric and give a ranking sum in the column R to compare their results in general. We observe that the integration of FoV90 + FoV120 improves CC but impairs NSS and AUC-J, the integration of FoV120 + FoV360 improves NSS but impairs KLD and CC. However, when we integrate three FoVs, it achieves the best KLD and keeps other metrics from decreasing too much. Therefore, we select $(\alpha, \beta, \gamma) = (0.2, 0.5, 0.3)$ as our best result as it has the highest total ranking and the scores of metrics are better than or equal to their medians.

D. Comparison with end-to-end structure

In our proposed framework, we separately fine tune a 2D saliency model with rectilinear images in three different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

TABLE IV
COMPARISON OF END-TO-END STRUCTURE AND OUR METHOD ON SALIENT360! 2017 DATASET [39] TEST SET. THE RESULTS IN BOLD INDICATE THE BEST SCORES ON EACH EVALUATION METRIC.

Method	KLD↓	CC↑	NSS↑	AUC-J↑
End-to-end structure	0.390	0.633	0.918	0.730
Our method	**0.363**	**0.662**	**0.978**	**0.747**

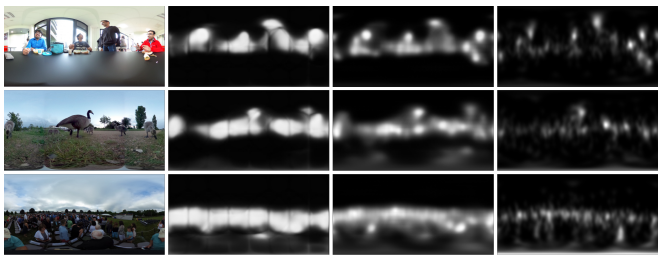

Fig. 11. Three examples of saliency maps generated from end-to-end structure, our framework, and groundtruth on Salient360! 2017 [39] test set. We can observe that saliency maps generated from end-to-end structure have obvious discontinuity and more blurry compared to that generated from our method.

FoVs, back-project the outputs into equirectangular format and combine them with linear integration. For improving computational efficiency, we also designed an end-to-end structure which is able to input rectilinear images of three FoVs in the same time, detect saliency features and output a 360° saliency map. The architecture of end-to-end structure is similar to Fig. 1, while the last sigmoid layers, which output the saliency map of three 2D models, are removed and a combinator with two convolutional layers and a sigmoid layer is attached to fuse saliency features detected from three FoVs. Hence, the output of each 2D model is 64-channel salient features. The salient features of FoV90 and FoV120 are back-projected to equirectangular format and concatenated with salient features of FoV360 to form 192-channel salient features. Then the combinator fuses these 192-channel features and outputs a 360° saliency map. Other implementation details are the same as that in our framework. This structure avoids the computational cost in densely cubic projection and linear integration of multiple FoVs. However, its performance, as shown in Table IV, is inferior to that of our framework proposed in this paper. We can see in Table IV that our method outperforms end-to-end structure by 7.4%, 4.6%, 6.5%, and 2.3% in KLD, CC, NSS, and AUC-J, respectively. Fig. 11 illustrates three examples of saliency maps generated from end-to-end structure, our proposed framework and corresponding groundtruth on the test set of dataset Salient360! 2017 [39]. We observe that there is obvious discontinuity in the saliency maps generated from end-to-end structure compared to that from our framework. It is due to cubic projection in end-to-end structure only rotates every 45° while that in our framework rotates every 10° in testing. From the results shown in Section V-B, smaller rotation angle achieves smoother saliency map in equirectangular format, but the more rectilinear images should be computed. Hence, the computational limitation of our 64GB computer only allows end-to-end structure to take rotation angle 45° as minimum. Besides, the computational limitation also constraints the input resolution of combinator in end-to-end structure to 256×512 as maximum, so that the output salient features should be downsampled as 128×128. It leads to end-to-end structure ignores important features and makes saliency maps more blurry as shown in Fig. 11. In contrast, our framework conserves the original resolution 256×256 of the output from 2D models, and generates the 360° saliency map in 512×1024, where the length and width is twice of that of the output from 2D models.

TABLE V
COMPARISON ON THE TEST SET OF SALIENT360! 2017 DATASET [39]. THE RESULTS IN BOLD AND BLUE COLOR RESPECTIVELY INDICATE THE BEST AND THE SECOND-BEST SCORES ON EACH EVALUATION METRIC.

Method	KLD↓	CC↑	NSS↑	AUC-J↑
Maugey *et al.* [43]	0.585	0.448	0.506	0.644
Zhang *et al.*[58]	-	0.409	0.699	0.659
SalNet360 [51]	0.458	0.548	0.755	0.701
SalGAN [28]	1.236	0.452	0.810	0.708
Startsev *et al.* [42]	0.42	0.62	0.81	0.72
GBVS360 [45]	0.698	0.527	0.851	0.714
BMS360 [45]	0.599	0.554	0.936	0.736
SalGAN&FSM [19]	0.896	0.512	0.910	0.723
Zhu *et al.* [46]	0.481	0.532	0.918	0.734
Ling *et al.* [50]	0.477	0.550	0.939	0.736
SalGAN360 [53]	0.431	0.659	0.971	0.746
Our method	**0.363**	**0.662**	**0.978**	**0.747**

TABLE VI
RESULTS ON THE TEST SET OF SALIENT360! 2018 BENCHMARK [65]. THE RESULTS IN BOLD AND BLUE COLOR RESPECTIVELY INDICATE THE BEST AND THE SECOND-BEST SCORES ON EACH EVALUATION METRIC.

Method	KLD↓	CC↑	SIM↑	NSS↑	AUC-J↑
SJTU model	1.238	0.520	0.573	1.397	0.820
Wuhan University	0.899	0.607	0.612	1.617	0.822
SalGAN360 [53]	0.739	0.642	0.635	1.585	0.820
Our method	**0.726**	**0.653**	**0.644**	**1.646**	**0.829**

TABLE VII
COMPARISON ON THE TEST SET OF SALIENCY IN VR - STANDING DATASET [41]. THE RESULTS IN BOLD AND BLUE COLOR RESPECTIVELY INDICATE THE BEST AND THE SECOND-BEST SCORES ON EACH EVALUATION METRIC.

Method	KLD↓	CC↑	NSS↑	AUC-J↑
Startsev *et al.* [42]	5.666	0.431	1.148	0.754
SalNet360[51]	5.849	0.390	1.200	0.772
SalGAN[28]	5.280	0.361	1.236	0.783
SalGAN&FSM [19]	5.333	0.375	1.286	0.794
SalGAN360 [53]	4.659	0.488	1.530	0.829
Our method	**4.325**	**0.507**	**1.551**	**0.830**

E. Comparison with the State-of-the-arts

We take our model of three-FoV SalGAN learned with adaptive weighting loss function and stretch weighted maps as our best-performed model and compare it with the state-of-the-arts in Salient360! 2017, Salient360! 2018, and Saliency in VR datasets.

Table V compares our model with 11 saliency prediction models trained and validated with 40 images and tested on 25-image test set in Salient360! 2017 dataset. SalGAN is compared here to present the performance of 2D model used in 360° images without any modification. The other 7 models listed in Table V, which are Maugey *et al.* [43], SalNet360[51], GBVS360 [45], BMS360 [45], Startsev *et al.* [42], Ling *et*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

al. [50] and Zhu *et al.* [46], are the participants of the Grand Challenge Salient360! ICME2017. The performance is evaluated by the organizers of the challenge. We follow the suggestion from MIT Saliency Benchmark [66] to rank all the models according to NSS scores. Our model outperforms the others on the 4 evaluation metrics, especially on KLD, which surpasses the second-best model by 15.1%. On the other metrics, our model is 0.4%, 0.7%, and 0.1% better than the second best on CC, NSS, and AUC-J, respectively.

Table VI compares our model with the other three models participated in ICME2018 Grand Challenge. Our model and theirs are all trained and validated on 85 images in the Salient360! 2018 dataset, tested on 26-image test set which the groundtruth is kept secretly, and submitted to the benchmark built by the challenge organizers. Thus, all the performance is evaluated by them with their private test dataset. Our model achieves the best result on all the 5 indexes by 1.8%, 1.7%, 1.4%, 1.8%, 0.9% on KLD, CC, SIM, NSS, AUC-J, respectively compared to the second-best model.

Table VII presents the performance of our model and other 5 state of the art methods in Saliency in VR dataset. All the models listed here use the same training dataset as those listed in Table V and tested on all 22 images in Saliency in VR dataset (*i.e.,* all 22 images serve as test set). Our model surpasses all of these 5 models on KLD, CC, and NSS by 7.7%, 3.9%, 1.4% respectively compared to the second-best method.

F. Qualitative comparison

Fig. 12 and Fig. 13 illustrate the qualitative results obtained by our model and other state-of-the-art models on Salient360! 2017 evaluation set, and Saliency in VR dataset. We can see that [19] overlooks abundant important features, because it directly downsamples the whole high resolution 360° images into small size 192×256 for 2D DCNN saliency model. [42] and [51] use viewport-based method which allow them be able to extract fine features in high resolution 360° images. However, their methods still have serious defects on the border of cube faces maps. Our model is capable of capturing both fine and coarse features via multi-FoVs method, and concentrating on the salient parts in equator area via fine tuning with adaptive weighting loss function attached with stretch weighted maps. Compared to the groundtruth, the saliency maps of our model are still blurry on the salient regions. It should be improved on how to detect finer yet more accurate salient features in high resolution 360° images in the future work.

G. Comparison with SalGAN360

From Table V, Table VI, and Table VII, we can see that the performance of our method are close to that of our previous work SalGAN360 [53]. It is due to the identical 2D saliency model, and the similar training and testing strategy of cubic projection on the limited scale of training datasets (*i.e.,* 40 images in Salient360! 2017 and 85 images in Salient360! 2018). The improvements from SalGAN360 to our method are three-FoV integration and adaptive weighting loss function with stretch weighted maps. In order to validate their contributions, we take a deeper look into the results of SalGAN360 and our method in every image in test sets of three datasets. Fig. 14, Fig. 15, and Fig. 16 illustrate the scores of KLD, CC, and NSS of each test image in dataset Salient360! 2017, Salient360! 2018, and Saliency in VR. As SalGAN360 integrates FoV90 and FoV360, these figures also show our method of FoV90 and FoV360 integration as "Our method (2FoVs)". Thus, the contribution of adaptive weighting loss function with stretch weighted maps can be seen by comparing SalGAN360 with our method (2FoVs), and the contribution of three FoVs integration can be seen by comparing our method (2FoVs) and our method. In Fig. 14, we can see that adaptive weighting losses with stretch weighted maps drastically decrease KLD scores in most of test images and three FoVs integration further deceases KLD in some amount, while they decrease CC scores in the same time but three FoVs integration further increases CC. Adaptive weighting losses with stretch weighted maps also improve NSS, while three FoVs integration improves NSS in some images but reduces NSS in other images. In Fig. 15 and Fig. 16, adaptive weighting loss function with stretch weighted maps and three FoVs integration decrease KLD, increase CC and NSS in most of test images. By observing Fig. 14, Fig. 15 and Fig. 16, we can conclude that adaptive weighting loss function with stretch weighted maps generally improves KLD, CC and NSS in three datasets but the only exception of CC in the Salient360! 2017. Besides, three FoVs integration also generally improves KLD, CC and NSS in three datasets but the only exception of NSS in the Salient360! 2017. The small scale of Salient360! 2017 (30 images to train, 10 images to validate and 25 images to test) could limit the effect of adaptive weighting losses and the integration of three FoVs by improving some evaluation metrics while sacrificing other metrics in the same time. In Fig. 15 and Fig. 16, we can see more consistent improvements on adaptive weighting losses and the integration of three FoVs in a larger scale dataset Salient360! 2018 (65 images to train, 20 images to validate and 26 images to test).

H. Discussion

In Table IV, we prove that the end-to-end structure is not superior to linear combination of multi-FoV saliency maps due to the discontinuity between viewports and computational limitation. Inspired by [67], which detected salient objects in 2D images with an adjustable combination weight of two models via reinforcement learning, and [68], which estimated saliency maps of 360° images with a fusion DCNN to combine saliency from different rotation viewports, it is promising for us to establish a more efficient method than linear combination for future work. On the basis of saliency prediction in 360° images, we can also follow the work in [14], [69] to extend our framework for 360° videos to predict head movement in the forthcoming frames.

VI. Conclusion

In this paper, we proposed a novel saliency prediction model for 360° images which considers the entire 360° image in three FoVs and its diverse viewport images. Our experiments

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

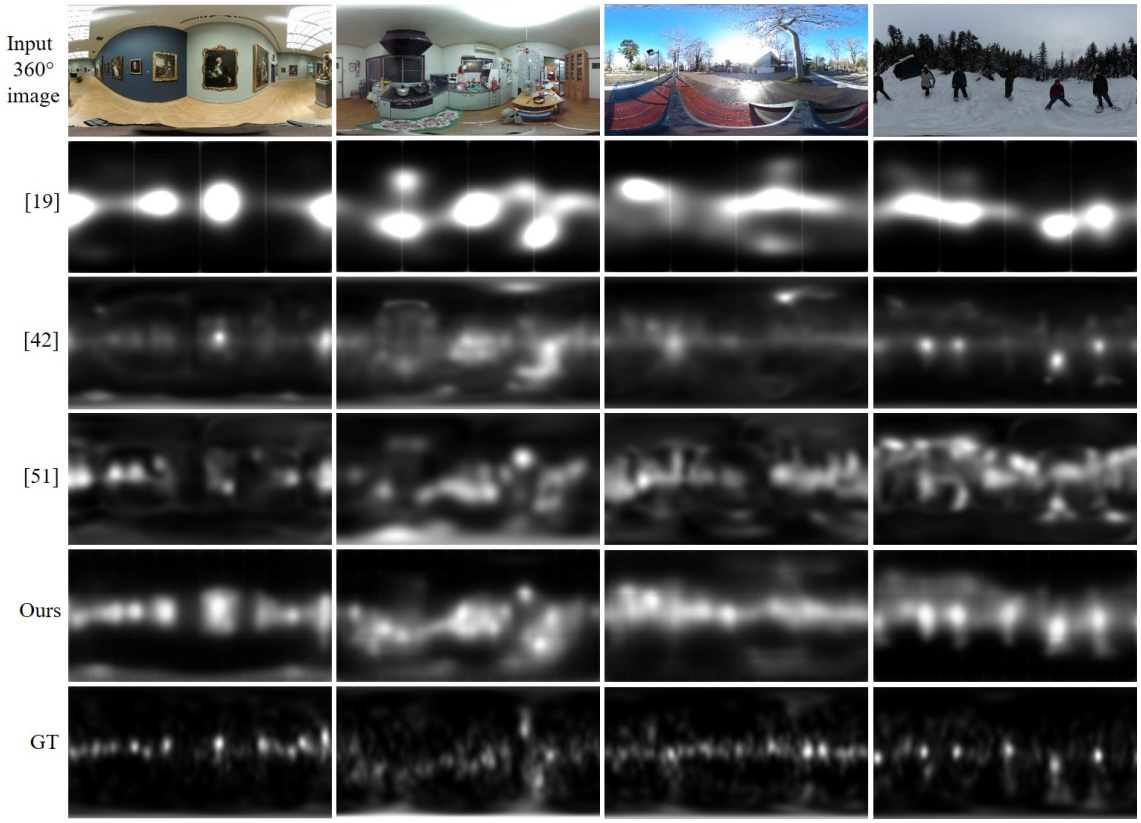

Fig. 12. Qualitative results and comparison with other state of the art models on Salinet360! 2017 [39] test set.

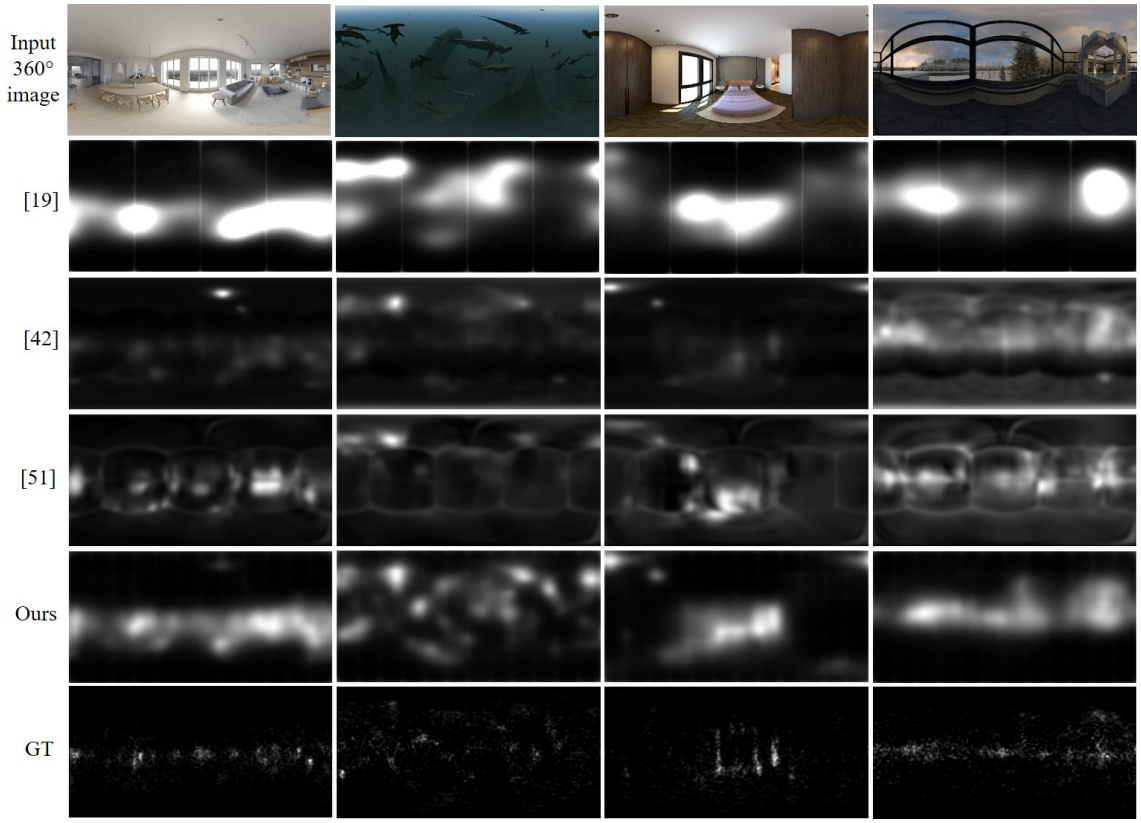

Fig. 13. Qualitative results and comparison with other state of the art models on Saliency in VR dataset [41] test set. The view angle of Gaussian blur is set to 1° in this dataset, so that the saliency regions in the groundtruth are much smaller than that in Salinet360! 2017 [39] dataset (view angle is 3.34°).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3003642, IEEE Transactions on Multimedia

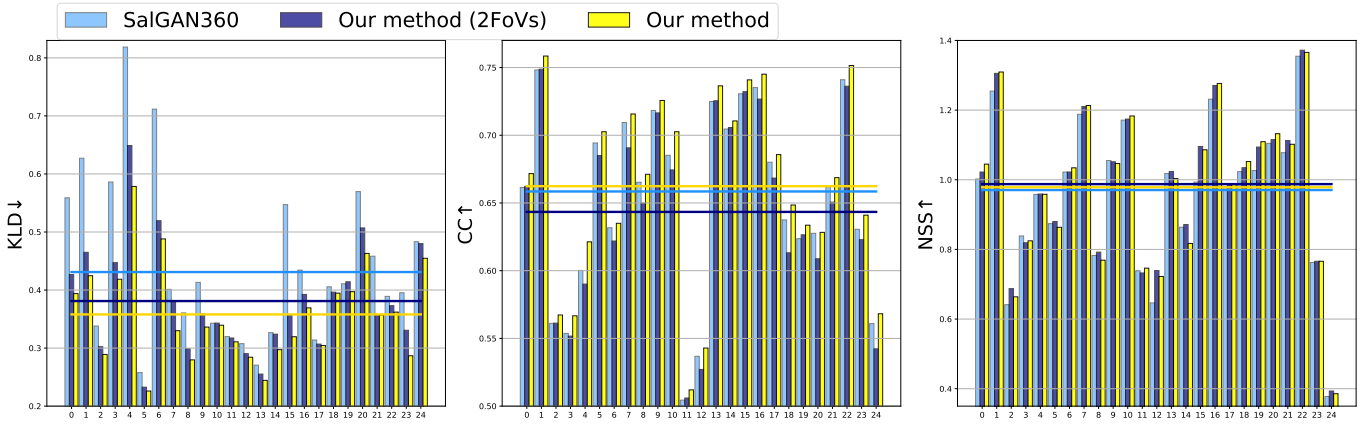

Fig. 14. Comparison with SalGAN360 in every test image in Salient360! 2017 [39] test set.

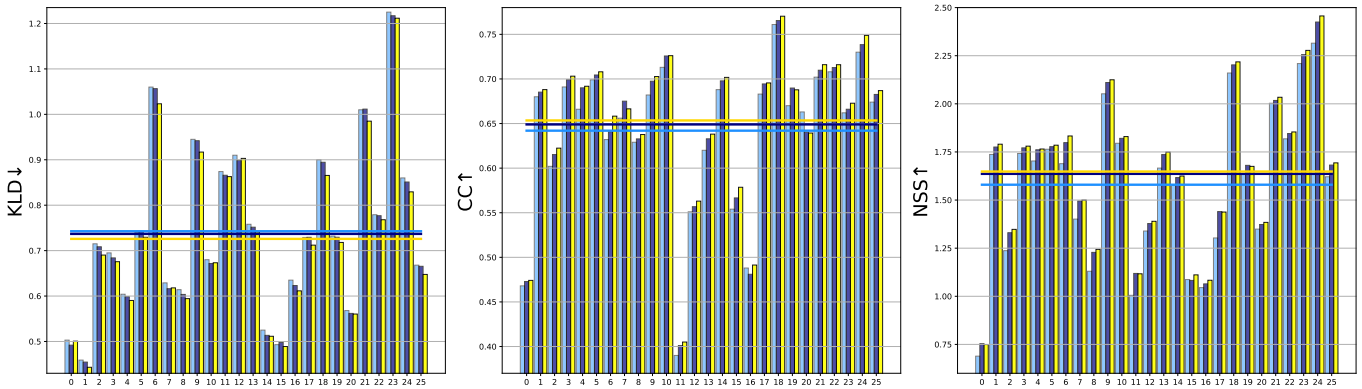

Fig. 15. Comparison with SalGAN360 in every test image in Salient360! 2018 [40] test set.

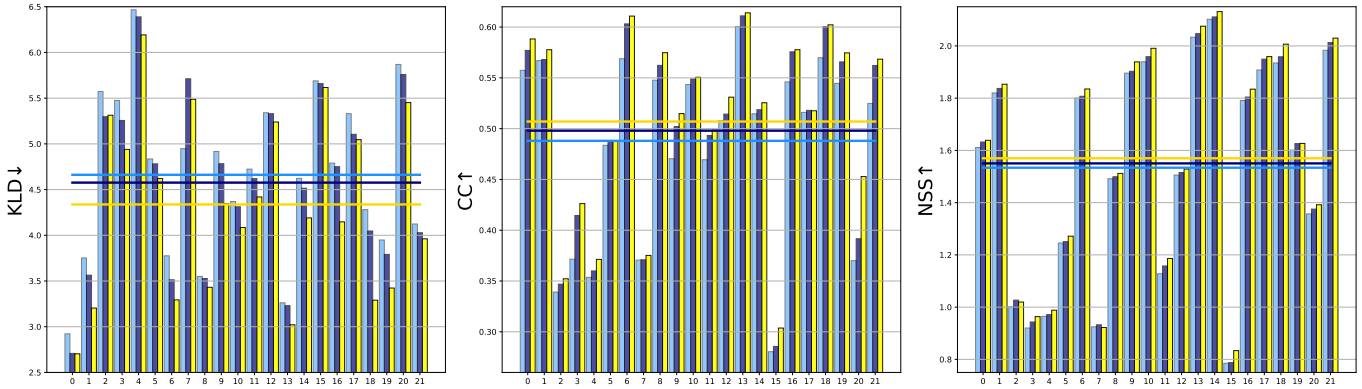

Fig. 16. Comparison with SalGAN360 in every test image in Saliency in VR [41] test set.

showed that the FoV size $90° \times 90°$ is not always the best choice for viewport based 360° saliency method in line with the fact that different FoV size has different effect on evaluation metrics. A better result on multiple evaluation metrics can be achieved in the same time by integrating saliency maps of FoV size $90° \times 90°$, $120° \times 120°$, and $360° \times 180°$. The other novelty of the proposal is an adaptive weighting loss function which dynamically balances the contribution of each evaluation metrics. It prevents the weights from manually tuning, and also outperforms the fixed weighting solutions. The same idea can be potentially applied on other combination loss functions. Stretch weighted maps were also introduced to lessen the impact of stretched regions in equirectangular images. Each component in our method has been validated to demonstrate its effectiveness. We also showed that our method outperforms other state of the arts on three public available datasets. Compared the saliency maps predicted from our model to the groundtruth, our model is not concentrated enough on the salient regions. In our future work, we plan to improve the model on detecting finer yet more accurate features in high resolution 360° images.

REFERENCES

[1] M. Tang, J. Wen, Y. Zhang, J. Gu, P. Junker, B. Guo, G. Jhao, Z. Zhu, and Y. Han, "A universal optical flow based real-time low-latency

omnidirectional stereo video system," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 957–972, April 2019.

[2] J. Li, Z. Wang, S. Lai, Y. Zhai, and M. Zhang, "Parallax-tolerant image stitching based on robust elastic warping," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1672–1687, July 2018.

[3] N. Li, Y. Xu, and C. Wang, "Quasi-homography warps in image stitching," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1365–1375, June 2018.

[4] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1819–1831, Sep. 2016.

[5] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "Mec-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.

[6] C. Fu, L. Wan, T. Wong, and C. Leung, "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 634–644, June 2009.

[7] Heung-Yeung Shum, King-To Ng, and Shing-Chow Chan, "A virtual reality system using the concentric mosaic: construction, rendering, and data compression," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 85–95, Feb 2005.

[8] Y. Su and K. Grauman, "Learning compressible 360° video isomers," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7824–7833.

[9] M. Xu, C. Li, Z. Chen, Z. Wang, and Z. Guan, "Assessing visual quality of omnidirectional videos," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2018.

[10] C. Li, M. Xu, X. Du, and Z. Wang, "Bridge the gap between vqa and human behavior on omnidirectional video: A large-scale dataset and a deep learning model," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18, 2018, pp. 932–940.

[11] S. Ling, G. Cheung, and P. Le Callet, "No-reference quality assessment for stitched panoramic images using convolutional sparse coding and compound feature selection," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, July 2018, pp. 1–6.

[12] C. Ozcinar, J. Cabrera, and A. Smolic, "Visual attention-aware omnidirectional video streaming using optimal tiles for virtual reality," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 217–230, March 2019.

[13] C. Ozcinar and A. Smolic, "Visual attention in omnidirectional video for virtual reality applications," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2018, pp. 1–6.

[14] C. Fan, S. Yen, C. Huang, and C. Hsu, "Optimizing fixation prediction using recurrent neural networks for 360° video streaming in head-mounted virtual reality," *IEEE Transactions on Multimedia*, pp. 1–1, 2019.

[15] Y. Rai, P. Le Callet, and P. Guillotel, "Which saliency weighting for omni directional image quality assessment?" in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017, pp. 1–6.

[16] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[17] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze prediction in dynamic 360° immersive videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 5333–5342.

[18] E. Upenik, M. Řeřábek, and T. Ebrahimi, "Testbed for subjective evaluation of omnidirectional visual content," in *2016 Picture Coding Symposium (PCS)*, Dec 2016, pp. 1–5.

[19] A. De Abreu, C. Ozcinar, and A. Smolic, "Look around you: Saliency maps for omnidirectional images in VR applications," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017, pp. 1–6.

[20] Y. Rai, J. Gutiérrez, and P. Le Callet, "A dataset of head and eye movements for 360 degree images," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 205–210.

[21] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, Jan 2013.

[22] W. Wang, J. Shen, F. Guo, M. Cheng, and A. Borji, "Revisiting video saliency: A large-scale benchmark and a new model," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 4894–4903.

[23] A. Borji, "Saliency prediction in the deep learning era: An empirical investigation," *CoRR*, vol. abs/1810.03716, 2018. [Online]. Available: http://arxiv.org/abs/1810.03716

[24] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, 2006, pp. 545–552.

[25] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 153–160.

[26] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 96–105, Jan 2013.

[27] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 262–270.

[28] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. a. Giro-i Nieto, "SalGAN: Visual saliency prediction with generative adversarial networks," in *arXiv*, January 2017.

[29] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, Oct 2018.

[30] S. Jia, "EML-NET: an expandable multi-layer network for saliency prediction," *CoRR*, vol. abs/1805.01047, 2018. [Online]. Available: http://arxiv.org/abs/1805.01047

[31] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1688–1698, July 2018.

[32] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 2106–2113.

[33] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1072–1080.

[34] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 280–287.

[35] A. Smolic and H. Kimata, "Applications and requirements for 3DAV," 2003.

[36] H. Strasburger, I. Rentschler, and M. Jüttner, "Peripheral vision and pattern recognition: A review," *Journal of Vision*, vol. 11, no. 5, pp. 13–13, 12 2011.

[37] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *CoRR*, vol. abs/1610.01708, 2016. [Online]. Available: http://arxiv.org/abs/1610.01708

[38] K. Zhang and Z. Chen, "Video saliency prediction based on spatial-temporal two-stream network," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2018.

[39] University of Nantes, "Salient360!: Visual attention modeling for 360 images grand challeng," in the IEEE International Conference on Multimedia and Expo (ICME), 2017.

[40] J. Gutiérrez-Cillán, E. J. David, Y. Rai, and P. L. Callet, "Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images," *Sig. Proc.: Image Comm.*, vol. 69, pp. 35–42, 2018.

[41] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1633–1642, April 2018.

[42] M. Startsev and M. Dorr, "360-aware saliency estimation with conventional image saliency predictors," *Signal Processing: Image Communication*, vol. 69, pp. 43 – 52, 2018.

[43] T. Maugey, O. Le Meur, and Z. Liu, "Saliency-based navigation in omnidirectional image," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, Oct 2017, pp. 1–6.

[44] F. Battisti, S. Baldoni, M. Brizzi, and M. Carli, "A feature-based approach for saliency estimation of omni-directional images," *Signal Processing: Image Communication*, vol. 69, pp. 53 – 59, 2018.

[45] P. Lebreton and A. Raake, "GBVS360, BMS360, prosal: Extending existing saliency prediction models from 2d to omnidirectional images," *Signal Processing: Image Communication*, vol. 69, pp. 69 – 78, 2018.

[46] Y. Zhu, G. Zhai, and X. Min, "The prediction of head and eye movement for 360 degree images," *Signal Processing: Image Communication*, vol. 69, pp. 15 – 25, 2018.

[47] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 4–4, 11 2007.

[48] P.-H. Tseng, R. Carmi, I. G. M. Cameron, D. P. Munoz, and L. Itti, "Quantifying center bias of observers in free viewing of dynamic natural scenes," *Journal of Vision*, vol. 9, no. 7, pp. 4–4, 07 2009.

[49] Y. Fang, X. Zhang, and N. Imamoglu, "A novel superpixel-based saliency detection model for 360-degree images," *Signal Processing: Image Communication*, vol. 69, pp. 1 – 7, 2018.

[50] J. Ling, K. Zhang, Y. Zhang, D. Yang, and Z. Chen, "A saliency prediction model on 360 degree images using color dictionary based sparse representation," *Signal Processing: Image Communication*, vol. 69, pp. 60 – 68, 2018.

[51] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic, "SalNet360: Saliency maps for omni-directional images with CNN," *Signal Processing: Image Communication*, vol. 69, pp. 26 – 34, 2018.

[52] J. Pan, E. Sayrol, X. Giro-I-Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 598–606.

[53] F. Chao, L. Zhang, W. Hamidouche, and O. Deforges, "SalGAN360: Visual saliency prediction on 360 degree images with generative adversarial networks," in *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2018, pp. 01–04.

[54] H. Cheng, C. Chao, J. Dong, H. Wen, T. Liu, and M. Sun, "Cube padding for weakly-supervised saliency prediction in 360° videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1420–1429.

[55] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360° videos," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham: Springer International Publishing, 2017, pp. 154–171.

[56] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv e-prints*, p. arXiv:1409.1556, Sep 2014.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[58] Z. Zhang, Y. Xu, J. Yu, and S. Gao, "Saliency detection in 360° videos," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[59] "HTC VIVE Specs," https://www.vive.com/us/product/vive-virtual-reality-system/, accessed: 2020-06-17.

[60] "Oculus Rift," https://www.oculus.com/rift/, accessed: 2020-06-17.

[61] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 740–757, March 2019.

[62] C. Carlson, "How I made wine glasses from sunflowers," http://blog.wolfram.com/2011/07/28/how-i-made-wine-glasses-from-sunflowers/, accessed: 2020-06-17.

[63] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *European Conference on Computer Vision (ECCV)*, Sep. 2018.

[64] M. Eder and J.-M. Frahm, "Convolutions on spherical images," in *CVPR Workshops*, 2019.

[65] J. Gutiérrez, E. J. David, A. Coutrot, M. P. D. Silva, and P. L. Callet, "Introducing UN salient360! benchmark: A platform for evaluating visual attention models for 360° contents," *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–3, 2018.

[66] Z. Bylinskii, T. Judd, F. Durand, A. Oliva, and A. Torralba, "MIT saliency benchmark," http://saliency.mit.edu/.

[67] "Quality-aware dual-modal saliency detection via deep reinforcement learning," *Signal Processing: Image Communication*, vol. 75, pp. 158 – 167, 2019.

[68] I. Djemai, S. A. Fezza, W. Hamidouche, and O. Deforges, "Extending 2D Saliency Models for Head Movement Prediction in 360-degree Images using CNN-based Fusion," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, May 2020.

[69] A. Nguyen, Z. Yan, and K. Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction." New York, NY, USA: Association for Computing Machinery, 2018.