Panoramic Vision Transformer for Saliency Detection in 360° Videos

Heeseung Yun, Sehun Lee, and Gunhee Kim

Seoul National University, Seoul, Korea
{heeseung.yun, shlee}@vision.snu.ac.kr, gunhee@snu.ac.kr
https://github.com/hs-yn/PAVER

Abstract. 360° video saliency detection is one of the challenging benchmarks for 360° video understanding since non-negligible distortion and discontinuity occur in the projection of any format of 360° videos, and capture-worthy viewpoint in the omnidirectional sphere is ambiguous by nature. We present a new framework named **Pa**noramic **V**ision Transform**er** (PAVER). We design the encoder using Vision Transformer with deformable convolution, which enables us not only to plug pretrained models from normal videos into our architecture without additional modules or finetuning but also to perform geometric approximation only once, unlike previous deep CNN-based approaches. Thanks to its powerful encoder, PAVER can learn the saliency from three simple relative relations among local patch features, outperforming state-of-the-art models for the Wild360 benchmark by large margins without supervision or auxiliary information like class activation. We demonstrate the utility of our saliency prediction model with the omnidirectional video quality assessment task in VQA-ODV, where we consistently improve performance without any form of supervision, including head movement.

Keywords: 360° videos, saliency detection, vision transformer

1 Introduction

360° video understanding is critical for providing intelligent systems with omnidirectional perception. Simultaneous view in every direction helps agents better react in challenging environments for indoor navigation [1], autonomous driving [9, 52], and drone navigation [7], to name a few. Also, virtual reality and 360° action cameras have pervaded entertainment applications.

Visual saliency prediction is one of the representative benchmarks for 360° video understanding. It can filter irrelevant or redundant information in panoramic views, and thus promote summarization of 360° videos or dynamic rendering of virtual reality panorama. Unlike saliency detection in normal field-of-view (NFoV) imagery that often aims to distinguish salient foreground from background, saliency prediction in 360° videos stems from a simple yet nontrivial question: which direction to watch if you were in the scene? Foreground objects may not always be of interest, and salient direction is subjective and depends on

context. Hence, 360° saliency prediction has often been interpreted as automated cinematography [39, 12], highlight detection [54, 28], and attention tracking [24].

There are a few challenges in predicting visual saliency in 360° videos. First, in any format of 360° videos (*e.g.*, equirectangular, cubemap [22]), a non-negligible proportion of distortion and discontinuity hinders the accurate processing of omnidirectional view. Thus, it is nearly impossible to directly leverage models learned from normal videos at no cost. Some previous works [28, 12, 66] explicitly project a number of NFoV images from a panorama frame to estimate salient viewpoints. However, this may not be scalable in terms of both space and time since an order of magnitude larger number of NFoV images (*e.g.*, 81 for [28]) should be processed per single panorama frame. In another line of works, transferrable architectures are proposed to utilize pretrained knowledge from the NFoV domain [37, 38, 18]. They can process 360° input without modification but at the cost of geometric error or additional modules for finetuning.

Second, ambiguity is another vital issue for saliency prediction in 360° videos. While previous works define saliency as intensity and orientation [25], self-information [4], and anomaly [45], there is no definitive answer for which constitutes capture-worthiness or saliency in 360° videos. A widely accepted tool to interpret saliency is class activation maps (CAM) [65] in both NFoV domain [59, 58, 51, 32] and 360° domain [11]. Although CAM can readily capture objects in the scene, it depends on the class labels of the reference dataset and is challenging to integrate with self-supervised pretraining that has no labels. As 360° saliency detection has been usually addressed under minimal or zero supervision, some other works resolve ambiguity by leveraging additional information like the coordinates of target objects [24] or reference NFoV videos of the same topic [39, 54, 28].

To address these issues, we propose a novel framework for 360° video saliency prediction named **Pa**noramic **V**ision Transform**er** (PAVER). It is equipped with two components: a deformation-aware omnidirectional encoder and a consistency-oriented saliency map decoder. First, our encoder adopts deformable convolution [14] to represent a 360° video as a set of small patches with local tangent projection for minimal geometric error. It can replace the NFoV projection that previous works often use with 60× less geometric error at negligible computation overhead. Then, we use the Vision Transformer [17] to remove the need for additional finetuning to transfer pretrained weights from the NFoV domain. As a result, the geometric approximation happens *only once* in our framework, unlike previous deep CNN-based approaches that perform at every layer, relieving the model of layerwise geometric error accumulation [38]. Our work is the first attempt to exploit the vision transformer to process 360° imagery.

Second, for our decoder to determine capture-worthy context on panoramic videos without supervision or additional information, we decompose the saliency into three relative relationships of the local patch features from its surrounding contexts. If the context of a local patch diverges from the overall representation of the video (*local saliency*), the patch can be deemed anomalous and is usually worth noticing. Moreover, if the spatial and temporal neighbors of a

patch are capture-worthy (*spatial & temporal saliency*), the patch should also be capture-worthy. By enforcing this simple yet straightforward objective in the feature dimension, we outperform the previous state-of-the-art model by 23% in the Wild360 dataset [11].Also, we leverage this saliency prediction for omni-directional video quality assessment for virtual reality (VR) in VQA-ODV [30], which is crucial for user experience in VR.

In conclusion, we summarize our main contribution as follows.

1. Our PAVER framework is the first attempt to adopt the Vision Transformer [17] to encode the omnidirectional imagery. Along with deformable convolution [14], our encoder alleviates geometric projection errors with no additional module and trivially processes panoramic videos in various formats by transferring the weights learned from the normal video datasets.
2. Thanks to our powerful encoder, we demonstrate that it is sufficient for 360° video saliency prediction to simply learn from relative relations among local patch features, outperforming state-of-the-art models for the Wild360 benchmark [11] by large margins with no additional annotations.
3. For the applicability of PAVER, we show that PAVER can consistently improve the performance of omnidirectional video quality assessment in the VQA-ODV [30] benchmark with no human supervision like head movement.

2   Related Work

Panoramic Video Processing. Efficient and accurate processing of 360° images or videos has been studied much. One of the most popular approaches is to project panorama into a set of normal field-of-view (NFoV) videos. Despite its simplicity, it has been effective in various tasks like vision-and-language navigation [66], language-guided view grounding [12], and 360° video summarization [28]. However, it requires explicit projection of up to 81 NFoV images per panoramic frame, which is less scalable in both space and time.

For the processing of spherical inputs, some prior works suggest designated architectures with spherical correlation [13], spherical convolution with spectral smoothness [19], or operations on unstructured grids [26]. Although they ensure favorable mathematical properties like rotational equivariance, they cannot be transferred from model weights trained with large sets of normal images or videos. Another compelling direction is to use transferrable architectures combined with geometric adaptation modules with finetuning [37, 38], polyhedral approximation [29], or both [60, 18]. However, additional modules for geometric alignment may be detrimental to latency, either in the training or inference step.

We use a transformer architecture where geometric approximation happens only once at the beginning, unlike previous approaches that perform in every layer. Pretrained weights from the NFoV domain are transferrable to our approach without pretraining for geometric adaptation. In addition, our approach is format-independent; not only the equirectangular format, but our model can also compute the cubemap or other formats without explicit conversion.

Visual Saliency Detection. Saliency detection has been a longstanding problem in computer vision research. In order to identify visual saliency, previous approaches utilize intensity and orientation with respect to stimulus [25], self-information maximization [4], intrinsic and extrinsic anomaly [45], and self-resemblance [35]. More recent work focuses on class activation maps [65], where high activation value of a certain class implies saliency. CAM-based methods are widely accepted for multi-source saliency detection [59], weakly supervised semantic segmentation [58], and object localization [51, 32]. We point our readers to a survey of visual saliency detection [43] for further details.

On the other hand, saliency prediction in 360° videos needs to identify capture-worthy viewpoints within the omnidirectional surroundings. Given a number of possible candidate viewpoints, it aims at providing plausible viewpoints or a heatmap as if someone is watching the scene. Thus, the saliency in 360° videos is often ambiguous and depends on subjective context. To resolve this, some works exploit NFoV videos as exemplars of capture-worthiness [39, 28, 54]. Other works leverage human supervision of saliency maps [62] or object tracking information [24] for training. Some recent works also utilize learned class activation maps [11] or natural language narratives [12]. Our approach is also in line with [11, 12] in that we do not require explicit supervision for training. One key difference is that we do not rely on additional information like CAM or narratives for training. Instead, we enforce local saliency and spatiotemporal saliency in the feature map of the local patch context.

Vision Transformers. Vision Transformers [17] have been drawing much attention for large-scale visual understanding since they reported impressive performance in image classification [55, 42], object detection and semantic segmentation [31, 50]. Recently, vision transformers have been adapted to broader domains, including point cloud [63] and video understanding [2, 61]. Another line of works focuses on transferring pretrained knowledge of vision transformers in an unsupervised or semi-supervised manner by leveraging self-distillation [8], semantics reallocation [21], seed propagation [36], and normalized cut [47].

Our work is the first to adopt the vision transformer to process omnidirectional inputs. Closest to our approach is Yun *et al.* [57], which utilizes the transformer for indoor semantic segmentation with monocular 360° images. However, Yun *et al.* do not take into account 360° format when processing images but instead discard the near-polar region where the spherical distortion is severe. On the other hand, our approach can process the whole panorama in a format-aware manner without discarding any parts, while being applicable to both 360° images and videos with only trivial overhead.

3   Approach

We present a model named **Pa**noramic **V**ision Transform**er** (PAVER) for saliency detection in 360° videos, as illustrated in Fig. 1. Given a 360° input video in equirectangular format $V = \{v^t\}_{t=1}^{T} \in \mathbb{R}^{T \times 3 \times H \times W}$ with T frames, our objective is to compute their saliency maps, *i.e.*, $\{\hat{y}^t\}_{t=1}^{T} \in \mathbb{R}^{T \times H \times W}$. In Sec. 3.1, we
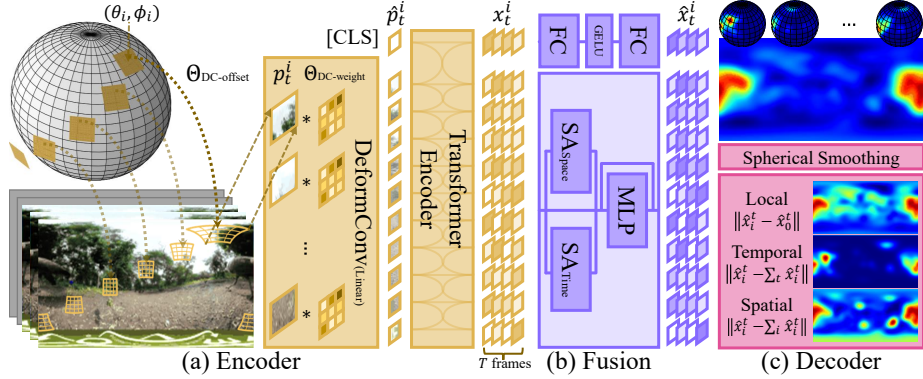
Fig. 1: Model architecture of **PAVER** (Panoramic Vision Transformer).

explain the Transformer encoder for encoding local patches with minimal geometric error (Fig. 1-(a)). Then, we deal with the spatiotemporal fusion module for learning consistent local and global features in Sec. 3.2 (Fig. 1-(b)). Finally, we account for the saliency map decoder with spherical smoothing and learning objectives in Sec. 3.3–3.4. Commonly used variables are described as follows:

S	Length of a tangential patch
W, H	Resolution of 360° video input (*e.g.*, $W = 448, H = 224$)
w, h	Number of patches along width and height, *i.e.*, $w = \frac{W}{S}, h = \frac{H}{S}$
N	Number of flattened patches, *i.e.*, $N = w \times h$
C	Number of channels per feature (*e.g.*, 768 for ViT-B/16 [17])

3.1   The Encoder for 360° Videos

Local Patch Projection. Naïve projection of local patches in vision transformers [17] cannot handle distortion and discontinuity in 360 domains. Hence, we leverage deformable convolution [14], which can process freeform deformation of convolution kernels with marginal computation overhead. For each 360° frame $v^t \in \mathbb{R}^{3 \times H \times W}$, we first compute tangential patches with a size of $S \times S$, namely $\{p_i^t\}_{i=1}^N$ where $N = wh = \frac{W}{S} \times \frac{H}{S}$. Then, we linearly project patches to obtain $\{\hat{p}_i^t\}_{i=1}^N$ where $\hat{p}_i^t \in \mathbb{R}^C$, using deformable convolution with fixed offsets:

$$\hat{p}^t = \text{Conv}(p^t) = \text{DeformConv}(v^t; \Theta_{\text{DC-weight}}, \Theta_{\text{DC-offset}}), \quad (1)$$

$$\Theta_{\text{DC-offset}}(\theta_i, \phi_i) = f_{\text{Sph} \to \text{ER}}(f_{\text{3D} \to \text{Sph}}(\mathcal{P} \times R(\theta_i, \phi_i) / \|\mathcal{P} \times R(\theta_i, \phi_i)\|_2)), \quad (2)$$

$$R(\theta_i, \phi_i) = \begin{pmatrix} \cos\phi_i\cos\theta_i & -\cos\phi_i\sin\theta_i & \sin\theta_i \\ \sin\theta_i & \cos\theta_i & 0 \\ \sin\phi_i\cos\theta_i & -\sin\phi_i\sin\theta_i & \cos\phi_i \end{pmatrix}, \quad (3)$$

where $(\theta_i, \phi_i) \in (0, 2\pi) \times (-\pi/2, \pi/2)$ is the longitude and latitude of the center of the i-th patch, and $\mathcal{P} \in \mathbb{R}^{S \times S \times 3}$ indicates sampled 3D points from an $S \times S$ patch on $z = 1$ plane. $f_{\text{3D} \to \text{Sph}}$ is the conversion from the 3D cartesian coordinates to the spherical coordinates, while $f_{\text{Sph} \to \text{ER}}$ is that from the spherical

to the 2D equirectangular coordinates. In a nutshell, we compute the offset for DeformConv by rotating the reference patch $\mathcal{P}$ by θ_i and ϕ_i and projecting it onto an equirectangular plane. Please refer to the Appendix for more details.

In Eq. 1, the weights for linear projection $\Theta_{\text{DC-weight}}$ can be transferred from pretrained Vision Transformers [17, 42, 8, 10] without additional tuning, as $\Theta_{\text{DC-offset}}$ locally projects curved surfaces to flat NFoV images. Unlike deformable convolutional networks [14], the offset $\Theta_{\text{DC-offset}}$ is computed only once and fixed throughout the training. That is, throughout our architecture, we apply geometric approximation only here for local tangential patches. Another benefit is that we can adapt any format other than equirectangular (*e.g.*, cubemap) without an explicit conversion process as long as the offset $\Theta_{\text{DC-offset}}$ is computed from the conversion formula between different formats. Considering that recently more videos are using the equiangular cubemap format for better resolution, our approach can represent the video as-is, regardless of the formats.

Transformer Encoder. We prefix N patches $\{\hat{p}_i^t\}_{i=1}^N$ with a learnable vector $\hat{p}_0^t$ (*i.e.*, [CLS] token) and feed them to the transformer encoder:

$$\{x_i^t\}_{i=0}^N = \text{Transformer}(\{\hat{p}_i^t\}_{i=0}^N; \Theta_{\text{ViT}}), \tag{4}$$

where $\{x_i^t\}_{i=0}^N$ is the output features of the patches. As in the local patch projection of Eq. 1, pretrained weights on the NFoV domain (*e.g.*, ViT [17] trained with ImageNet-21k [15]) can be reused for Θ_{ViT} without finetuning or even additional modules. Unless mentioned otherwise, we use vanilla vision transformers with the ViT-B/16 weights [17]. Refer to Table 2-(b) later for the experiments with other transformer variants. More details are deferred to Appendix.

3.2   Spatiotemporal Fusion

Since the encoded features $\{x_i^t\}_{i=0}^N$ are obtained from each patch separately, we now train a spatiotemporal fusion module so that the local features are smoothly aligned with respect to space and time. We decouple global context x_0^t from the local patch context $\{x_i^t\}_{i=1}^N$ and model them separately. We observe that training both global and local context with identical weight is detrimental to performance, which is further discussed in Sec. 4.2.

Global Context. [CLS] tokens in Transformers are generally used as an input to a classification head to predict the output. Thus, we regard $x_0^t \in \mathbb{R}^C$, the output of transformer for the [CLS] input token, as encapsulating the global context of scenery in a sense. Finally, we encode global context $\hat{x}_0^t \in \mathbb{R}^C$ via a simple multilayer perceptron (MLP):

$$\hat{x}_0^t = \text{MLP}_{\text{G}}(x_0^t; \Theta_{\text{G}}), \tag{5}$$

where MLP refers to two fully-connected layers with GELU activation [23].

Local Context. To encode temporal information in the local patch context, we use features from all T frames *i.e.*, $X = \{\{x_i^t\}_{i=1}^N\}_{t=1}^T \in \mathbb{R}^{T \times N \times C}$. We extend the vanilla transformer encoder with multi-head self-attention [44] temporally:

$$\hat{X} = X' + \text{MLP}_{\text{L}}(X'; \Theta_{\text{L}}), \text{ where } X' = X + \frac{\text{SA}(X; \Theta_{\text{S}}) + \text{SA}(X^T; \Theta_{\text{T}})^T}{2}. \tag{6}$$

MLP_L indicates an MLP with residual connection, and SA respectively denotes multi-head self-attention along the temporal axis (T) and spatial axis (N). X^T is a transpose between the temporal and spatial axis, i.e., $X^T \in \mathbb{R}^{N \times T \times C}$. By averaging two multi-head self-attentions, local patches have similar feature representation with their spatial and temporal neighbors altogether.

3.3   The Decoder for Saliency Map

Using both global and local contexts ($\{\hat{x}_0^t\}_{t=1}^T$ and $\{\{\hat{x}_i^t\}_{i=1}^N\}_{t=1}^T$), we first compute the saliency score of each local patch. We decompose the saliency into three terms: local, temporal, and spatial saliency. Instead of directly optimizing saliency scores, we measure relative relations between each local context feature and its neighbors and optimize them on the feature level.

First, the local saliency measures how much a local patch deviates from the global context. If the distance between the local patch and global context is large, it can be deemed as an anomalous patch, which may be worth viewing. Likewise, temporal and spatial saliency reflects how much a local patch deviates from the temporal or spatial mean of its neighbors. With more distance between, the context of the patch would more differ from those of its neighborhoods. Finally, our saliency score is computed as follows, where $\alpha = \beta = \gamma = 1$ for simplicity:

$$y_i^t = \alpha \left\| \hat{x}_i^t - \hat{x}_0^t \right\|_2^2 + \beta \left\| \hat{x}_i^t - \frac{1}{T} \sum_{t=1}^T \hat{x}_i^t \right\|_2^2 + \gamma \left\| \hat{x}_i^t - \frac{1}{N} \sum_{i=1}^N \hat{x}_i^t \right\|_2^2. \qquad (7)$$

Spherical Gaussian Smoothing. To upsample from $\mathbb{R}^{h \times w}$ to $\mathbb{R}^{H \times W}$ while observing the spherical structure, we apply spherical Gaussian smoothing [16]. The scalar saliency score $\hat{y}^t$ is obtained as

$$\hat{y}_{ij}^t(\theta_i, \phi_i, \psi_j) = y_i^t \times \cos \phi_i \times \frac{a}{\sinh a} e^{a \cos \psi_j}, \text{ where } a = \frac{W^2}{4\pi^2 \sigma^2}. \qquad (8)$$

ψ_j denotes how much the j-th pixel deviates from (θ_i, ϕ_i), and σ is the standard deviation of Gaussian smoothing. Note that spherical Gaussian smoothing is only applied for evaluation purposes.

3.4   Learning Objectives

Without any ground truth label or additional information, we train the model by ensuring spatiotemporal consistency while maintaining the global context.

Temporal Consistency Loss. It is natural for two adjacent frames to display similar saliency values and feature distributions. We take into account two neighboring frames, i.e., $t+1$ and $t-1$ for the t-th frame:

$$\mathcal{L}_T = \frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N \left\| \hat{x}_i^t - \frac{\hat{x}_i^{t+1} + \hat{x}_i^{t-1}}{2} \right\|_2^2. \qquad (9)$$

Spatial Consistency Loss. Adjacent patches should retain similar saliency scores and feature distributions. We use the geodesic distance between patches to reflect the spherical structure.

$$\mathcal{L}_S = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} \left\| \hat{x}_i^t - \sum_{j=1}^{N} \frac{\gamma_i \delta_{ij} \hat{x}_j^t}{g_{ij}} \right\|_2^2, \tag{10}$$

where $g_{ij} = ||(\theta_i, \phi_i) - (\theta_j, \phi_j)||_g$ is the geodesic distance between the i-th and j-th patch, and $\delta_{ij} = 1$ when $0 < g_{ij} < \epsilon$ for some ϵ. γ_i is a scaling factor such that $\sum_{j=1}^{N} \frac{\gamma_i \delta_{ij}}{g_{ij}} = 1$. The idea of Eq. 10 is that the similarity of patches within a certain threshold of ϵ should be inversely proportional to their geodesic distance.

Global Context Loss. To train the MLP_G for global context, we encourage all T frames in a video to have a similar global context:

$$\mathcal{L}_G = \frac{1}{T} \sum_{t=1}^{T} \left\| \hat{x}_0^t - \frac{1}{T} \sum_{s=1}^{T} \hat{x}_0^s \right\|_2^2. \tag{11}$$

To sum up, our loss function is defined as follows:

$$\mathcal{L}_{\text{total}} = \lambda_T \mathcal{L}_T + \lambda_S \mathcal{L}_S + \lambda_G \mathcal{L}_G. \tag{12}$$

Training. We end-to-end train our model with a batch size of 1 and $T = 5$ frames per input. We fix the encoder weight with the pretrained weight of the Vision Transformer (ViT-B/16) [17]. We optimize with Adam optimizer [27] with a learning rate of 2e-7 for five epochs. For hyperparameter, we use $\lambda_T = 20, \lambda_S = 0.5, \lambda_G = 0.1$. Please refer to the Appendix for more details.

4   Experiments

For evaluation, we perform experiments on two benchmark tasks with Wild360 [11] and VQA-ODV [30] datasets. First, we evaluate our target task, saliency prediction on 360° videos, on Wild360 [11] as one of the most popular datasets. Second, we apply our approach to visual quality assessment on 360° videos on VQA-ODV [30]. The quality assessment of capture-worthy viewports is important in omnidirectional videos, and the models for this task usually require annotations from headgears or eye-trackers as well as human supervision of subjective assessment. If we can replace such expensive annotations with visual saliency maps, omnidirectional video quality assessment can become scalable without human supervision. We thus evaluate how much PAVER can improve the omnidirectional video quality assessment with no such annotations.

4.1   Experiment Setting

Datasets. Wild360 [11] is composed of 85 video clips about natural scenery, and split into 60 clips for training and 25 for test. Human annotated ground-truth

saliency heatmaps are provided only for test split. VQA-ODV [30] consists of 540 impaired videos from 60 lossless reference videos. Nine types of impairment are applied to each reference video with varying degrees of compression levels and projection types (ERP, RCMP, TSP). The quality of each impaired video is scored by 20 subjects wearing a head-mounted display, and annotated with the head movement (HM) and eye movement (EM) of a subject.

Evaluation Metrics. We use the standard measures of the two benchmarks. For Wild360, we report three metrics for saliency detection: AUC-Judd [34], AUC-Borji [3], and Linear Correlation Coefficient (CC). AUC-Judd computes the true positive and false positive rate of the saliency map. AUC-Borji randomly samples pixels to calculate false positive rates of these pixels. CC measures the linear relationship between the proposed saliency map and ground truth. We regard CC as the most important metric as recommended by [6]. Please refer to [5] for more details.

For VQA-ODV, differential mean opinion score (DMOS) quantifies the quality of omnidirectional videos perceived by the viewers. Common objective metrics for visual quality assessment like structural similarity (SSIM) [48] and peak signal-to-noise ratio (PSNR) are not necessarily proportional to actual human perception. Hence, Pearson correlation coefficient (PCC), Spearman rank correlation coefficient (SRCC), root mean squared error (RMSE) and mean absolute error (MAE) between the DMOS and target metrics are utilized as the quantitative metrics, which we report.

Baselines. First, we compare our PAVER model against competitive baselines for predicting 360° video saliency based on optical flow [49], gradient flow [46], generative adversarial networks [33], and class activation map with optical flow [11]. We also report the performance of unsupervised saliency detection and unsupervised object discovery models based on vision transformers, including TS-CAM [21], DINO [8], LOST [36] and TokenCut [47]. For a fair comparison with our approach, we use the identical local patch projection module in Sec. 3.1 and the spherical Gaussian smoothing module in Sec. 3.3.

For the ablation study, we also report five variants of our approach. PAVER (Cartesian) replaces all 360° aware components in our model with normal field-of-view (NFoV) equivalents. PAVER (NoGlobal) is the model without the $\mathrm{MLP_G}$ projection of global context. PAVER (NoLocal) replaces the local context spatiotemporal fusion module with a simple MLP. PAVER (NoDecoupled) encodes both global and local contexts together in a single transformer encoder. PAVER (ScoreLoss) is trained with the spatiotemporal score consistency (*i.e.*, the sparse saliency map Y) instead of the feature map consistency.

For VQA-ODV, we report the performance of PSNR, WS-PSNR [40], and S-PSNR [53] with different weight conditions: uniform, random, our saliency map, and reversed saliency map. Here the primary comparison is PSNR variants between our saliency map and the ones with human head movement supervision since PSNR metrics weighted with human head and eye movement supervision are considerably better in quality assessment [30]. More details of baseline models' configuration can be found in Appendix.

Table 1: Comparison of saliency prediction accuracy on the Wild360 dataset [11].

|	CC	AUC-J	AUC-B
Motion Magnitude [49]	0.288	0.687	0.642
ConsistentVideoSal [46]	0.085	0.547	0.532
SalGAN [33]	0.312	0.717	0.692
Equirectangular [11]	0.337	0.839	0.783
CubePad (Static) [11]	0.448	0.881	0.852
CubePad (CLSTM) [11]	0.420	0.898	0.859
TS-CAM [21]	0.414	0.831	0.802
DINO [8]	0.406	0.850	0.831
LOST [36]	0.444	0.809	0.786
TokenCut [47]	0.500	0.841	0.815
PAVER (NoGlobal)	0.376	0.814	0.797
PAVER (NoDecoupled)	0.492	0.881	0.860
PAVER (Cartesian)	0.549	0.898	0.875
PAVER (NoLocal)	0.561	0.895	0.873
PAVER (ScoreLoss)	0.575	0.906	0.883
PAVER	**0.616**	**0.923**	**0.899**

(a) Latitude

(b) Longitude
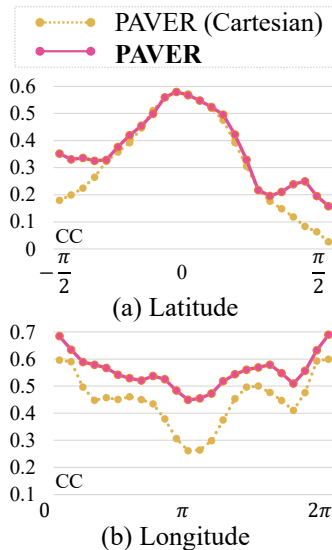
Fig. 2: Influence of distortion on Wild360 test split.

4.2 Results on Saliency Detection

Comparison with the Prior Arts. Table 1 compares the performance with prior arts on the Wild360 dataset. PAVER outperforms previous state-of-the-art models by large margins: +11.6%p (*i.e.*, 23%) in CC, +2.5%p in AUC-J, and +4.0%p in AUC-B. We achieve notable improvement especially in correlation coefficient (CC). That is, evaluating the relative relations among local patches is sufficient to achieve competent performance in the Wild360 dataset. Our approach is also time-efficient in that we do not require time-consuming computation for additional information like optical flow; for example, CubePad [11] requires 19 seconds per frame pair for flow computation [49].

Moreover, PAVER consistently achieves better performance compared to the baselines that use the vision transformer as the encoder backbone (*i.e.*, TS-CAM, DINO, LOST, and TokenCut). It is worth noticing that merely applying pretrained weights of the vision transformers does not guarantee superior performance. For instance, TS-CAM or DINO show slightly worse performance than the best performing non-ViT model. Although they are competent in detecting larger foreground objects, 360° videos usually contain multiple small objects of potential interest. As in Fig. 3-(a), the baselines with transformer encoders relatively fall short when multiple salient objects are worth viewing in the scene.

Ablation Studies. The last six rows in Table 1 compare the PAVER variants. First of all, performance drops significantly when the global context is poor, as PAVER without the global context encoder (*i.e.*, $\mathrm{MLP_G}$) plummets by 24%p in CC. PAVER without the decoupled global-local context encoder decreases CC by 12.4%p, which suggests that independent parameter update for

Table 2: (a) Performance comparison with different saliency score compositions, where L, T, and S denotes Local, Temporal, and Spatial saliency score, respectively. (b) Influence of different backbone architectures, pretrained weights, and resolution in our encoder on the Wild360 Dataset [11].

(a)						(b)				
L	T	S	CC	AUC-J	AUC-B	Backbone	Res.	CC	AUC-J	AUC-B
		✓	0.276	0.760	0.731	TimeSformer (T=16) [2]	224	0.465	0.863	0.843
✓			0.548	0.897	0.875	DINO (B/16) [8]	224	0.524	0.893	0.871
		✓	0.552	0.894	0.872	DINO (B/8) [8]	224	0.563	0.905	0.881
	✓	✓	0.557	0.899	0.876	ViT-Ti/16 [41]	224	0.540	0.904	0.881
✓		✓	0.585	0.907	0.884	ViT-B/16 [17]	384	0.567	0.893	0.871
✓	✓	✓	**0.616**	**0.923**	**0.899**	ViT-B/16 [17]	224	**0.616**	**0.923**	**0.899**

the global context encoder is essential for better performance. The performance of PAVER without 360° geometry-aware modules drops by 6.7%p, implying the importance of the 360° format awareness. Also, as in Fig. 2-(a), the gap between PAVER (Cartesian) and PAVER particularly widens in the near-polar region (*e.g.*, ×4 for $\theta = \frac{\pi}{2}$), where most of the distortion in 360° videos takes part. Replacing the spatiotemporal local context encoder with a simple MLP decreases CC by 5.5%p. The residual connections of both spatial and temporal self-attention are beneficial for generating saliency maps with high fidelity. If we enforce score-level consistency instead of feature-level consistency during training, the performance drops by 4.1%p in CC. In summary, all components in PAVER contribute to the full model's performance in their own ways.

Analysis on Saliency Score Composition. Table 2-(a) summarizes the influence of different saliency score components in Eq. 7. Computing saliency maps only with temporal saliency does not show acceptable performance. On the other hand, when the temporal saliency score is added to other sets of scores (*i.e.*, S→T+S, L+S→L+T+S), the performance consistently improves in all three metrics. Since the deviation of a local patch along the timeline is relatively small in magnitude, temporal saliency helps our saliency prediction update smoothly in time. Using only local or spatial saliency displays similar performance, but both of them fall short by 6–7%p when compared to the full model. These two saliency scores are complementary in that combining local and spatial saliency boosts performance by 3–4%p in CC.

Influence of Encoder Backbones. Table 2-(b) compares the performance of our model with different architectures, pretrained weights, and resolutions. First, we replace the video transformer in PAVER with TimeSformer [2]. It shows inferior performance mainly because the model is trained with a larger temporal hop size, as capturing subtle movement in scenery is essential for better saliency maps in the Wild360 dataset. When we use DINO [8], which requires even no labels as the backbone of our model, it shows better performance compared to the SOTA models. The model with a smaller patch size reports better CC by 4%p, which is presumably due to higher fidelity of the saliency map. Using ViT-

Table 3: Results of omnidirectional video quality assessment on VQA-ODV [30]. The better is higher PCC and SRCC or lower RMSE and MAE.

Weight	PSNR				WS-PSNR [40]				S-PSNR [53]			
	PCC	SRCC	RMSE	MAE	PCC	SRCC	RMSE	MAE	PCC	SRCC	RMSE	MAE
None	0.650	0.664	9.004	7.027	0.672	0.684	8.771	6.909	0.693	0.698	8.541	6.681
Random	0.650	0.663	9.004	7.027	0.672	0.684	8.771	6.909	0.693	0.698	8.540	6.680
Reverse	0.646	0.654	9.041	6.883	0.646	0.660	9.044	7.033	0.683	0.693	8.652	6.711
DINO [8]	0.657	0.677	8.934	7.105	0.674	0.690	8.747	7.033	0.699	0.710	8.468	6.665
PAVER	0.657	0.664	8.931	7.133	0.692	0.704	8.551	6.794	0.702	0.707	8.438	6.659
HM(Supervised)	0.733	0.726	8.054	6.479	0.731	0.722	8.086	6.565	0.736	0.741	8.022	6.305

Ti/16 [41], the performance drops by 6.6%p in exchange for 15× fewer model parameters. This is still better than existing SOTA models for all three metrics.

4.3 Omnidirectional Video Quality Assessment

Table 3 reports performance metrics between PSNR variants and DMOS. When using random weights for PSNR computation, the results are nearly identical to the ones with no weight assignment. That is, providing weights that are irrelevant to saliency does not improve the performance. If we use saliency maps from the PAVER as PSNR weights, the performance consistently improves, *e.g.*, 0.7%p in PSNR, 2.0%p in WS-PSNR, and 0.9%p in S-PSNR, respectively for the PCC metric. On the other hand, if we use reverse saliency maps as weights, the performance worsens compared to the PSNR variants without weight assignment. For instance, using correct saliency maps and reversed saliency maps displays a 4.6%p gap for the PCC metric in WS-PSNR. This implies that proper assignment of saliency maps helps solve the omnidirectional quality assessment.

Comparison with Head Movement Supervision. The last row of Table 3 reports the performance of PSNR variants with actual human head movement as weights. Compared to S-PSNR with head movement supervision, S-PSNR with our saliency maps shows 3.4%p lower performance in PCC. Unlike ground truth labels that require trackable headgears for annotation, our saliency map can be automatically computed using a couple of videos.

4.4 Qualitative Results.

Saliency Detection. Fig. 3-(a) compares some examples of saliency prediction by different methods. In general, our PAVER better captures contexts that are worth viewing. For example, our model places the highest score on the penguin moving towards the camera (a-1) or the polar bear instead of a jeep or a flag nearby (a-2). Also, compared to PAVER(Cartesian), our model can propose more sphere-aware saliency maps as in the second and third rows of Fig. 3-(a).

Fig 3-(b-c) compares the PAVER results according to the score decompositions. Local-only and space-only saliency maps look alike in that they both assign higher scores on anomalous patches. However, unlike space-only saliency maps,

(a) Model comparison

(b) Saliency score decomposition of PAVER (ViT-B/16 [17])

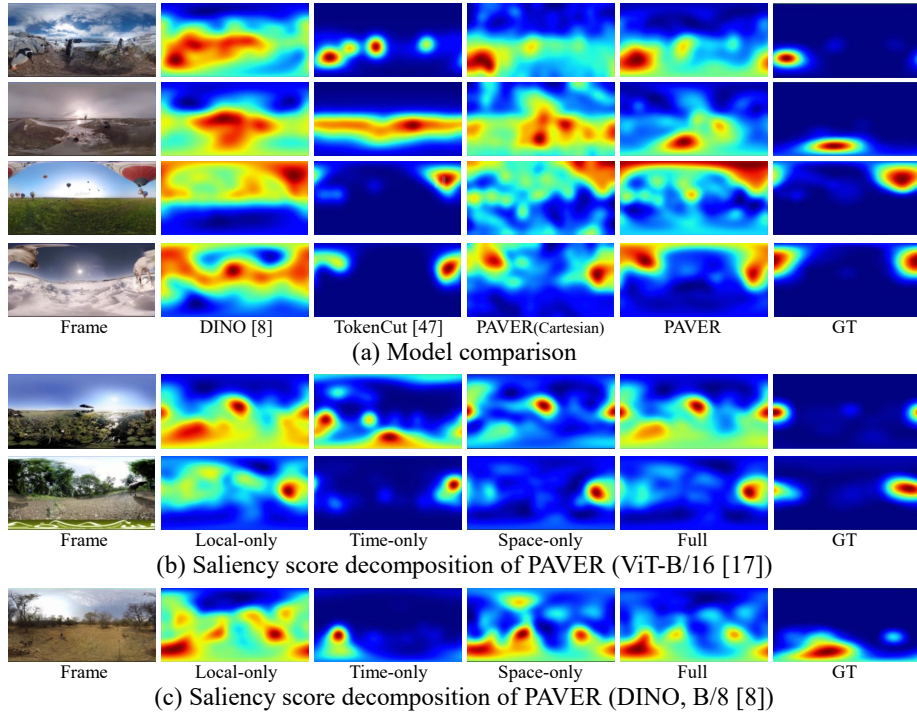(c) Saliency score decomposition of PAVER (DINO, B/8 [8])

Fig. 3: Qualitative comparison of saliency prediction on Wild360 [11].

local-only saliency maps tend to favor object-like patches for both supervised [17] and self-supervised pretraining [8]. Time-only saliency focuses on subtle movement in the scene, which helps generate smooth transitions of saliency maps. We present more qualitative examples in Appendix.

Quality Assessment. Fig. 4 illustrates our saliency prediction on the VQA-ODV dataset. Compared to head movement, our saliency maps are in line with what people think is worth viewing. Still, as in Fig. 4-(4), our model struggles in cases where the camera drastically moves.

360° Videos with Different Formats. Fig. 5 displays our saliency prediction on varying 360° video projection formats, including equirectangular (ERP), cubemap (RCMP), and truncated square pyramid projection (TSP). Using randomly sampled videos from the test split of Wild360, we convert them from ERP to RCMP or TSP using 360tools[1]. We replace $\Theta_{\text{DC-offset}}$ for ER with the offsets for each video format. Computation of $\Theta_{\text{DC-offset}}$ for different video formats can be found in Appendix. Our model can process different 360° formats without explicitly converting from one to another while returning highly consistent saliency maps regardless of the formats, especially for the ones with severe regional sampling discrepancy like TSP.
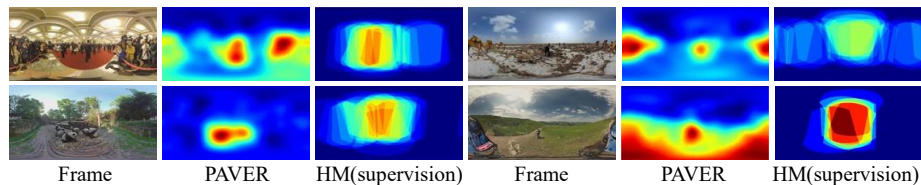
---

[1] https://github.com/Samsung/360tools

| Frame | PAVER | HM(supervision) | Frame | PAVER | HM(supervision) |

Fig. 4: Comparison of saliency and video quality prediction on VQA-ODV [30].

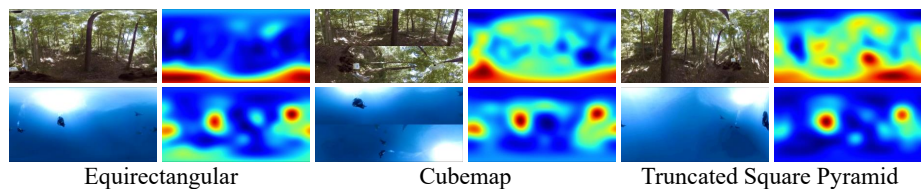| Equirectangular | Cubemap | Truncated Square Pyramid |

Fig. 5: Prediction comparison of different projection formats for 360° videos.

5   Conclusion

We proposed a new model for 360° video saliency prediction named PAVER. We regarded 360° videos as a set of patches using deformable convolution, which alleviates the need for layerwise geometric approximation unlike other CNN-based approaches. We adopted the vision transformer to encode omnidirectional imagery by reusing pretrained knowledge from normal videos with no need for complex adaptation. Even with three simple feature-wise consistency objectives, PAVER outperformed prior arts that use additional annotations or vision transformers in Wild360. We also achieved consistent improvement in omnidirectional video quality assessment in VQA-ODV. Last but not least, all computations in PAVER can be adapted for various 360° formats without explicit conversion.

There are multiple interesting future directions beyond this work. First, we can extend PAVER to be adaptable with multi-scale vision transformers such as Swin transformer [31], MViT [20], SegFormer [50], and SeTR [64]. Since they can be effective for fine-grained prediction, geometry-aware multi-scale encoding can be beneficial for a better understanding of omnidirectional imagery. Second, our work could be used as a generic omnidirectional encoder for various tasks like language-guided view grounding [12], embodied navigation [66], and 360° video question answering [56]. Another direction is to combine both 360° and NFoV inputs to train a unified architecture to leverage the complementary nature of the two formats.

References

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language Navigation: Interpreting Visually-grounded Navigation Instructions in Real Environments. In: CVPR (2018)
2. Bertasius, G., Wang, H., Torresani, L.: Is Space-Time Attention All You Need for Video Understanding? In: ICML (2021)
3. Borji, A., Tavakoli, H.R., Sihite, D.N., Itti, L.: Analysis of scores, datasets, and models in visual saliency prediction. In: ICCV (2013)
4. Bruce, N., Tsotsos, J.: Saliency based on information maximization. In: NIPS (2005)
5. Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., Torralba, A.: MIT Saliency Benchmark (2015)
6. Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., Durand, F.: What Do Different Evaluation Metrics Tell Us about Saliency Models? IEEE TPAMI (2018)
7. Caron, G., Morbidi, F.: Spherical Visual Gyroscope for Autonomous Robots Using the Mixture of Photometric Potentials. In: ICRA (2018)
8. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021)
9. Caruso, D., Engel, J., Cremers, D.: Large-scale Direct SLAM for Omnidirectional Cameras. In: IROS (2015)
10. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV (2021)
11. Cheng, H.T., Chao, C.H., Dong, J.D., Wen, H.K., Liu, T.L., Sun, M.: Cube padding for weakly-supervised saliency prediction in 360 videos. In: CVPR (2018)
12. Chou, S.H., Chen, Y.C., Zeng, K.H., Hu, H.N., Fu, J., Sun, M.: Self-view grounding given a narrated 360 video. In: AAAI (2018)
13. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical CNNs. In: ICLR (2018)
14. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable Convolutional Networks. In: ICCV (2017)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-scale Hierarchical Image Database. In: CVPR (2009)
16. Devaraju, B.: Understanding Filtering on the Sphere: Experiences from Filtering GRACE Data. Ph.D. Dissertation, Inst. Geodesy, Univ. Stuttgart (2015)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020)
18. Eder, M., Shvets, M., Lim, J., Frahm, J.M.: Tangent images for mitigating spherical distortion. In: CVPR (2020)
19. Esteves, C., Allen-Blanchette, C., Makadia, A., Daniilidis, K.: Learning so (3) equivariant representations with spherical cnns. In: ECCV (2018)
20. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale Vision Transformers. In: ICCV (2021)
21. Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., Zhou, B., Ts-cam, Q.Y.: Token Semantic Coupled Attention Map for Weakly Supervised Object Localization. In: ICCV (2021)
22. Greene, N.: Environment Mapping and Other Applications of World Projections. IEEE CGA (1986)
23. Hendrycks, D., Gimpel, K.: Gaussian Error Linear Units (GELUs). arXiv:1606.08415 (2016)

24. Hu, H.N., Lin, Y.C., Liu, M.Y., Cheng, H.T., Chang, Y.J., Sun, M.: Deep 360 Pilot: Learning a Deep Agent for Piloting through 360 Sports Videos. In: CVPR (2017)
25. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE TPAMI (1998)
26. Jiang, C.M., Huang, J., Kashinath, K., Marcus, P., Niessner, M., et al.: Spherical CNNs on Unstructured Grids. In: ICLR (2018)
27. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
28. Lee, S., Sung, J., Yu, Y., Kim, G.: A memory network approach for story-based temporal summarization of 360 videos. In: CVPR (2018)
29. Lee, Y., Jeong, J., Yun, J., Cho, W., Yoon, K.J.: Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images. In: CVPR (2019)
30. Li, C., Xu, M., Du, X., Wang, Z.: Bridge the gap between VQA and human behavior on omnidirectional video: A large-scale dataset and a deep learning model. In: ACMMM (2018)
31. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
32. Meng, M., Zhang, T., Tian, Q., Zhang, Y., Wu, F.: Foreground Activation Maps for Weakly Supervised Object Localization. In: ICCV (2021)
33. Pan, J., Ferrer, C.C., McGuinness, K., O'Connor, N.E., Torres, J., Sayrol, E., Giro-i Nieto, X.: Salgan: Visual saliency prediction with generative adversarial networks. arXiv:1701.01081 (2017)
34. Riche, N., Duvinage, M., Mancas, M., Gosselin, B., Dutoit, T.: Saliency and human fixations: State-of-the-art and study of comparison metrics. In: ICCV (2013)
35. Seo, H.J., Milanfar, P.: Nonparametric Bottom-up Saliency Detection by Self-resemblance. In: CVPRw (2009)
36. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., Ponce, J.: Localizing Objects with Self-Supervised Transformers and no Labels. In: BMVC (2021)
37. Su, Y.C., Grauman, K.: Learning spherical convolution for fast features from 360 imagery. NIPS (2017)
38. Su, Y.C., Grauman, K.: Kernel transformer networks for compact spherical convolution. In: CVPR (2019)
39. Su, Y.C., Jayaraman, D., Grauman, K.: Pano2Vid: Automatic Cinematography for Watching $360°$ Videos. In: ACCV (2016)
40. Sun, Y., Lu, A., Yu, L.: Weighted-to-spherically-uniform quality evaluation for omnidirectional video. SPL (2017)
41. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training Data-efficient Image Transformers & Distillation through Attention. In: ICML (2021)
42. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: ICCV (2021)
43. Ullah, I., Jian, M., Hussain, S., Guo, J., Yu, H., Wang, X., Yin, Y.: A brief survey of visual saliency detection. MTA (2020)
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
45. Wang, M., Konrad, J., Ishwar, P., Jing, K., Rowley, H.: Image saliency: From intrinsic to extrinsic context. In: CVPR (2011)
46. Wang, W., Shen, J., Shao, L.: Consistent video saliency using local gradient flow optimization and global refinement. TIP (2015)

47. Wang, Y., Shen, X., Hu, S., Yuan, Y., Crowley, J., Vaufreydaz, D.: Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut. In: CVPR (2022)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004)
49. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large displacement optical flow with deep matching. In: ICCV (2013)
50. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: NeurIPS (2021)
51. Xie, J., Luo, C., Zhu, X., Jin, Z., Lu, W., Shen, L.: Online Refinement of Low-level Feature Based Activation Map for Weakly Supervised Object Localization. In: ICCV (2021)
52. Yogamani, S., Hughes, C., Horgan, J., Sistu, G., Varley, P., O'Dea, D., Uricár, M., Milz, S., Simon, M., Amende, K., et al.: Woodscape: A Multi-task, Multi-camera Fisheye Dataset for Autonomous Driving. In: ICCV (2019)
53. Yu, M., Lakshman, H., Girod, B.: A framework to evaluate omnidirectional video coding schemes. In: ISMAR (2015)
54. Yu, Y., Lee, S., Na, J., Kang, J., Kim, G.: A deep ranking model for spatio-temporal highlight detection from a $360°$ video. In: AAAI (2018)
55. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. In: ICCV (2021)
56. Yun, H., Yu, Y., Yang, W., Lee, K., Kim, G.: Pano-AVQA: Grounded Audio-visual Question Answering on 360deg Videos. In: ICCV (2021)
57. Yun, I., Lee, H.J., Rhee, C.E.: Improving 360 Monocular Depth Estimation via Non-local Dense Prediction Transformer and Joint Supervised and Self-supervised Learning. In: AAAI (2022)
58. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L.: Joint Learning of Saliency Detection and Weakly Supervised Semantic Segmentation. In: ICCV (2019)
59. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L., Qian, M., Yu, Y.: Multi-source Weak Supervision for Saliency Detection. In: CVPR (2019)
60. Zhang, C., Liwicki, S., Smith, W., Cipolla, R.: Orientation-aware semantic segmentation on icosahedron spheres. In: ICCV (2019)
61. Zhang, Y., Li, X., Liu, C., Shuai, B., Zhu, Y., Brattoli, B., Chen, H., Marsic, I., Tighe, J.: Vidtr: Video transformer without convolutions. In: ICCV (2021)
62. Zhang, Z., Xu, Y., Yu, J., Gao, S.: Saliency detection in 360 videos. In: ECCV (2018)
63. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)
64. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In: CVPR (2021)
65. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. In: CVPR (2016)
66. Zhu, F., Zhu, Y., Chang, X., Liang, X.: Vision-language Navigation with Self-supervised Auxiliary Reasoning Tasks. In: CVPR (2020)