

Pathformer3D: A 3D Scanpath Transformer for 360° Images

Rong Quan¹, Yantao Lai^{1,2}, Mengyu Qiu², and Dong Liang^{2,3} (✉)

¹ College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, the Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education, Nanjing, 211106, China

`rongquan0806@gmail.com, yantaolai@nuaa.edu.cn`

² MIT Key Laboratory of Pattern Analysis and Machine Intelligence, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

`{qmengyu, liangdong}@nuaa.edu.cn`

³ Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen, China

Abstract. Scanpath prediction in 360° images can help realize rapid rendering and better user interaction in Virtual/Augmented Reality applications. However, existing scanpath prediction models for 360° images execute scanpath prediction on 2D equirectangular projection plane, which always result in big computation error owing to the 2D plane’s distortion and coordinate discontinuity. In this work, we perform scanpath prediction for 360° images in 3D spherical coordinate system and proposed a novel 3D scanpath Transformer named Pathformer3D. Specifically, a 3D Transformer encoder is first used to extract 3D contextual feature representation for the 360° image. Then, the contextual feature representation and historical fixation information are input into a Transformer decoder to output current time step’s fixation embedding, where the self-attention module is used to imitate the visual working memory mechanism of human visual system and directly model the time dependencies among the fixations. Finally, a 3D Gaussian distribution is learned from each fixation embedding, from which the fixation position can be sampled. Evaluation on four panoramic eye-tracking datasets demonstrates that Pathformer3D outperforms the current state-of-the-art methods. Code is available at <https://github.com/lisztzp/Pathformer3D>.

Keywords: Scanpath · 360° image · 3D Transformer

1 Introduction

Virtual reality (VR) and augmented reality (AR) [50] technologies have made significant advancements in recent years, providing the users immersive experiences. Meanwhile, understanding and imitating the way human beings explore the 360° images of virtual environments is becoming more and more important,

since it can help realize more practical and fast rendering and thus improve user interaction in the immersive environment [24, 51]. Scanpath prediction on 360° images refers to predict human’s gaze shift path when exploring 360° images.

Existing research on scanpath prediction has primarily focused on 2D images [4, 22, 42, 46, 53], while 360° images possess distinct characteristics in the VR/AR context. 360° images provide immersive interactive environments where users can change their perspectives by physically moving their heads, resulting in a wider distribution of fixations. 360° images encompass richer visual information that requires more time for users to process and absorb. Furthermore, the data structure of 360° images differs from that of 2D images, introducing new requirements for data processing and analysis. Therefore, existing 2D image scanpath prediction methods cannot be directly applied to predict scanpaths in 360° images.

Early methods [6, 43, 57] for predicting scanpaths in 360 images involved sampling fixations based on saliency information to obtain the entire scanpath. Subsequently, with the advancement of generative adversarial networks [2], some researchers [5, 35] have utilized generative networks to directly generate the entire path from the 360° image. These methods have shown preliminary results but overlook the modeling of time dependency between fixations, which has long been demonstrated to be a very important characteristic of human visual attention mechanism [22], thereby they often lead to unstable prediction results. Recently, some research works [18, 33, 38, 45] have recognized the need for a comprehensive treatment of the temporal dependencies in viewing behavior, and modeled them through recurrent neural networks (RNNs) or Markov chains. However, these methods integrate all historical fixation information into a single hidden unit, and generate the current fixation only from this hidden unit. In this case, the time dependent relationships among fixations can only be modeled indirectly and the influences from historical fixations will be greatly weakened. Besides, all above methods predict fixations on the 2D equirectangular projection of 360° image, which has the issue of coordinate discontinuity caused by longitude (where -180° and 180° represent the same fixation despite their significant numerical difference), and different degrees of distortion in different locations. Consequently, the fixation predicted on this 2D equirectangular projection will have a large margin of error.

To solve the above problems, we propose a novel scanpath prediction model for 360° images named Pathformer3D, which predicts fixations in 3D spherical coordinates of the 360° image and directly model the time dependent relationships among fixations, to mimic human viewing behavior in immersive environment more realistically. Specifically, given a 2D equirectangular projection of 360° image, we first transform it into 3D spherical coordinate system, and then exploit a spherical convolution to extract its visual features. After extracting the 3D visual features, we exploit a 3D Transformer encoder to learn each 360° image region a contextual feature representation, by considering their long range spatial dependencies between each other. Next, we exploit a Transformer Decoder to learn each time step’s fixation embedding directly from the whole image’s

visual feature and the historical fixations, where the self-attention module is used to mimic human visual working memory mechanism, which can model the time dependencies between the current fixation and all historical fixations. After obtaining each time step’s fixation embedding, we employ a 3D mixture density network to learn the 3D Gaussian distribution for each fixation, from which the fixation location is sampled. Using 3D Gaussian distribution to model the positional likelihood of fixations in 3D space takes into account the differences in scanpaths of different individuals, and can thus result in more robust and real scanpaths.

In conclusion, this work has the following contributions:

- We perform visual scanpath prediction for 360° images in a 3D spherical coordinate system for the first time, which can realize a more real imitation of human visual exploration process in immersive environment and simultaneously avoid the error caused by the distortion and positional discontinuity in 2D equirectangular projection of 360° image.
- We propose a novel scanpath prediction method named Pathformer3D, to learn contextual feature representation of the 360° images in a 3D spherical coordinate system and directly consider the influences from historical fixations on current fixation, which is more in line with the visual working memory mechanism of human vision system.
- The proposed method is comprehensively evaluated on four eye-tracking datasets of 360° images, consistently achieving state-of-the-art performance, demonstrating the effectiveness of our approach.

2 Related Works

Scanpath Prediction for 2D Images In recent years, numerous models have been proposed for predicting scanpaths in 2D images. According to the different sources of inspiration, they can be categorized into different classes.

Some of these models draw inspiration from neuroscience and vision science, often leveraging human gaze behavior to guide scanpath prediction. For instance, some researchers [17, 22, 46] have utilized Inhibit-of-Return(IOR) strategies to guide the generation of scanpaths. Additionally, Wang *et al.* [52] compute response maps of sparse coding filters applied to foveated images at each step and select new fixations from the residual perceptual information map(RPI) using the information maximization principle. Adeli *et al.* [1] first apply a retina transformation to input images, then compute corresponding priority maps and select the next fixation. Furthermore, some researchers [48, 56] attempt to generate new fixations by leveraging low-level semantic information.

Another subset of work is inspired by the statistical distribution characteristics of human scanpaths. Many researchers [7, 8, 30, 32, 34, 42, 53] in this group attempt to use image saliency information to predict new scanpaths. For example, VSPT [42] uses SalGAN [39] to extract saliency information and employs an autoregressive approach [21] to generate fixations. Others in this category adopt alternative methods, such as Sun *et al.* [47], who conduct super-Gaussian quality

analysis on image patches and select the k th fixation to be at the patch with the highest response to the k th super Gaussian component. Coutrot *et al.* [13] model scanpaths using image and task dependent hidden markov models(HMM). Clarke *et al.* [11] implement a Gaussian jump distribution, which is polynomially dependent on the previous fixed distribution.

In addition to the aforementioned models, there are also some deep learning models [5, 28, 55] that have achieved relatively good performance, despite their limited interpretability.

Scanpath Prediction for 360° Images Unlike 2D images, there are currently limited models specifically designed for scanpath prediction in 360° images. However, with the gradual development of VR/AR technologies, this field has attracted increasing attention. In this domain, SaltiNet [6] is the first proposed method, which uses saliency volume to process 360° images. PathGAN [5] adopts the method of Generative Adversarial [20] to generate scanpath of 360° image. Zhu *et al.* [58] proposed a new method to estimate the saliency of 360° images, and introduced mechanisms such as visual uncertainty and visual balance to predict the scanpath. The subsequent ScanGAN [35] is also a method that uses Generative Adversarial Networks [20] to generate scanpath. Different from PathGAN [5], ScanGAN [35] adds a soft-dtw loss function [15] to the generator to generate scanpath that are more similar to real scanpath. ScanDMM [45] is a method that uses Deep Markov model [27] to generate scanpath, which takes into account the past historical information.

All above methods predict fixations as well as scanpaths in 2D equirectangular projection of 360° image, which has the weaknesses of distortion and coordinate discontinuity and always introduce big prediction error. Different from them, we perform scanpath prediction for 360° images in 3D spherical coordinate system, which is more consistent with the real-world scenarios of human exploring immersive environments.

3 Approach

3.1 Problem Definition

A scanpath can be represented as a sequence of fixations $X = (p_1, p_2, \dots, p_T) \in \mathbb{R}^{T \times 3}$, where T represents the length of the scanpath. For a given 360° image, the task of a scanpath prediction model is to generate a scanpath \hat{X} that resembles a real human scanpath. We use a 3D Cartesian coordinate system to represent coordinates, where each fixation is given in the form of $p = (x, y, z)$, with x , y , and z being real numbers ranging from -1 to 1. This representation form helps address the issue of coordinate discontinuity.

3.2 Overview

The architecture of our model is shown in Fig. 1. For each input 360° image, we first transform it into 3D spherical coordinate system, and then extract the

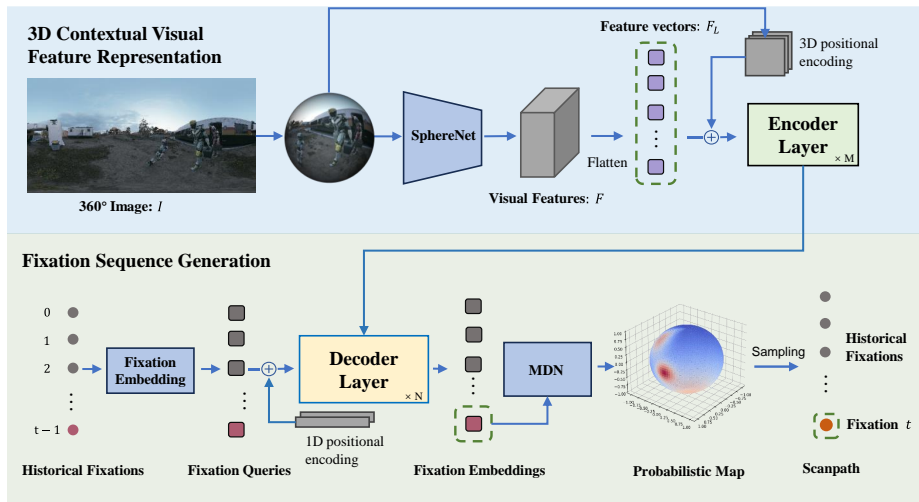


Fig. 1: Overall architecture of our methods. The Transformer Encoder is utilized to contextualize the feature extracted by the SphereNet [12]. The contextual feature representation and the historical fixations are fed into the Transformer Decoder to output fixation embeddings. Each fixation embedding is input into a 3D Mixture Density Network to output its 3D Gaussian distribution, from which the fixation is sampled.

image features using SphereNet [12]. Subsequently, we feed the extracted feature representation into a 3D Transformer Encoder to contextualize them. The Transformer Decoder takes historical fixation information and contextual feature representations as input and generates embedding information for new fixations in an autoregressive manner [21]. Then, a 3D Mixture Density Network (MDN) takes the embedding information of the fixations as input and outputs their 3D Gaussian position distributions. Finally, the fixations are sampled from the 3D Gaussian distributions. Next, we will introduce a detailed module-wise description of our model.

3.3 3D Contextual Visual Feature Representation

3D feature extraction. We perform scanpath prediction in 3D spherical coordinate system. Therefore, after projecting the 360° image into 3D spherical coordinate system, we utilize SphereNet [12] to extract its 3D feature representation. Specifically, for a 360° image I , after inputting it into SphereNet, we extract 3 different scales of feature maps and concatenate them along the channel axis, resulting in a feature representation F with dimensions of $C = 448, H = 128, W = 256$.

Before inputting into the 3D Transformer Encoder, we first execute a simple size transformation to F . Firstly, we execute average pooling to F in a convolutional kernel of size 8×8 , to reduce the size of the feature maps to $C = 448, H = 16, W = 32$. Next, we flatten the pooled feature maps, resulting

in a feature F_L with size of $L = 16 \times 32 = 512$, $C = 448$. This series of operations can be formulated as:

$$F = \text{SphereNet}(I, \theta) \quad (1)$$

$$F_L = \text{Flatten}(\text{AvgPool}(F)) \quad (2)$$

here, θ represents the parameters of SphereNet.

3D feature encoder. A linear embedding layer is first applied to map F_L into feature embeddings. Subsequently, the obtained feature embeddings are added to a 3D positional encoding and fed into $M = 4$ standard Transformer Encoder Layers to extract contextual feature representations. The above process can be expressed by the following formulas.

$$F'_L = \text{Embedding}(F_L) \quad (3)$$

$$P_0 = F'_L + E_{pos_3d} \quad E_{pos_3d} \in R^{L \times D} \quad (4)$$

$$P_i = \text{EncoderLayer}(P_{i-1}) \quad i = 1 \dots M \quad (5)$$

where D represents the feature embedding dimension of the 3D Transformer Encoder, M represents the number of EncoderLayer in the architecture, and P_i represents the output of the i -th EncoderLayer.

3.4 Fixation Sequence Generation

Fixation decoder. The contextual feature representation and historical fixation information are input into a Transformer decoder to output the fixation embedding. The fixation embedding is then fed into a 3D mixture density network to output the fixation's 3D Gaussian distribution, from which the fixation is sampled. We use autoregressive method [21] to predict one fixation each time and iteratively generate the whole fixation sequence.

On time step t , we initialize the fixation query vector q_t using fixation p_{t-1} of the previous time step. Specially, for the first fixation query q_1 , we initialize it using the center of 3D space, *i.e.*, $p_0 = (0, 0, 0)$. Subsequently, the initialized query vector q_t is concatenated with the historical query vectors to generate the joint query vector Q'_0 . After adding one-dimensional positional encoding information, Q'_0 becomes Q_0 . Finally, after passing through $N = 4$ standard Transformer Decoder Layers, we obtain the hidden states $Q_N = (Z_1, Z_2, \dots, Z_t)$ of the fixations. This process is defined as follows:

$$q_t = \text{Embedding}(p_{t-1}) \quad t \in [1, 2, \dots, T] \quad (6)$$

$$Q'_0 = [q_1; q_2; \dots; q_t] \quad (7)$$

$$Q_0 = Q'_0 + E_{pos_1d} \quad E_{pos_1d} \in R^{t \times D} \quad (8)$$

$$Q_i = \text{DecoderLayer}(Q_{i-1}, P_M) \quad i = 1 \dots N \quad (9)$$

where P_M represents the output of the Transformer Encoder, T represents the length of the predicted fixation sequence, p_{t-1} represents the fixation at the time

step $t - 1$, N represents the number of decoder layers. Q_i represents the output of the i -th DecoderLayer.

The fixation decoder process in the training process has some difference from that in the generation process. During the training phase, we input the fixations $(p_0, p_1, \dots, p_{T-1}) \in R^{T \times 3}$ into the Transformer Decoder, and it can predict the hidden states $Q_N = (Z_1, Z_2, \dots, Z_T)$ of the true scanpath $(p_1, p_2, \dots, p_T) \in R^{T \times 3}$ in a parallel manner, which is a parallel process. During the generation phase, we iteratively generate the hidden states of new coordinates Z_1, Z_2, \dots, Z_T in sequence starting from the initial coordinate p_0 , which is a sequential process.

Fixation generation. After obtaining Z_t , we then employ a 3D mixture density network to model the probability distribution of the fixations. The 3D mixture density network decodes the hidden state Z_t of the fixation at time step t into K sets of Gaussian kernel parameters. We use $\mu_t^i = (\mu_{x,t}^i, \mu_{y,t}^i, \mu_{z,t}^i)$, Σ_t^i , and π_t^i to represent the mean vector, covariance matrix, and weight of the i -th Gaussian kernel at time step t , respectively. These three parameters can be obtained through three separate MLP layers, which can be defined as:

$$\{\mu_t^i, \Sigma_t^i, \pi_t^i\} = f_{mdn}(Z_t; \theta_{mdn}) \quad i = 1 \dots K \quad (10)$$

where θ_{mdn} refers to the parameters of the mixture density network.

Next, the probability density function for a 3D Gaussian distribution $P(p_t)$ can be defined as follows:

$$\mathcal{N}(p_t | \mu_t^i, \Sigma_t^i) = \frac{\exp(-\frac{1}{2}(p_t - \mu_t^i)^\top (\Sigma_t^i)^{-1} (p_t - \mu_t^i))}{\sqrt{(2\pi)^3 |\Sigma_t^i|}} \quad (11)$$

$$P(p_t) = \sum_{i=1}^K \pi_t^i \mathcal{N}(p_t | \mu_t^i, \Sigma_t^i) \quad (12)$$

where the $|\Sigma_t^i|$ means the determinant of the covariance matrix Σ_t^i and the $\mathcal{N}(p_t | \mu_t^i, \Sigma_t^i)$ represents the Gaussian distribution function associated with the i -th 3D Gaussian function.

The prediction fixation \hat{p}_t is sampled probabilistically from a 3D surface, which can be expressed as:

$$\hat{p}_t \sim P(p_t) \quad (13)$$

here, it is worth noting that we do not simply sample the fixation with the highest probability, since such sampling strategy always make the model rapidly converge to the same fixation positions for all images. In our study, we generated a total of 32,768 fixation points along the latitude-longitude grid, with a resolution of 128×256 . We independently computed the probability for each fixation point. Finally, we randomly selected a fixation point based on its probability, with higher probabilities being more likely to be chosen.

3.5 Loss Function

During training, the objective is to minimize the discrepancy between the predicted distribution and the real human scanpath. To achieve this, we employ the negative log-likelihood loss function, which encourages higher probabilities for the ground truth fixations. The loss function can be defined as follows:

$$\mathcal{L} = -\frac{1}{T} \sum_{t=1}^T \log \left(\sum_{i=1}^K \pi_t^i \mathcal{N}(p_t^* | \mu_t^i, \Sigma_t^i) \right) \quad (14)$$

where p_t^* represents the real human fixation position of time step t .

4 Experiments

4.1 Datasets

We conducted experiments on four 360° image datasets: Sitzmann [44], Salient360! [43], AOI [54], and JUFÉ [19]. The Sitzmann dataset [44] comprises 22 images with a total of 1920 scanpaths. Only the training set of the Salient360! dataset [43] is available, which contains 85 images and 3036 scanpaths. The AOI dataset [54] consists of 600 high-resolution images and approximately 18,000 scanpaths. The JUFÉ dataset [19] includes 1032 images and 30,960 scanpaths. For training, 19 images from Sitzmann [44] and 60 images from Salient360! [43] were utilized, and all remaining data were used for validation. To increase the number of training sets, we rotated the image along with the corresponding scanpaths six times along the direction of longitude. We followed ScanDMM [45] to pre-process the gaze data of Sitzmann, Salient360!, and JUFÉ datasets, and obtain visual scanpaths containing 30, 25, and 15 fixations for Sitzmann, Salient360!, and JUFÉ. The AOI dataset has already been pre-processed, with an average scanpath length of approximately 8.

4.2 Evaluation Metrics

We employed six metrics to assess the performances of our model and other models, including Levenshtein Distance (LEV) [41], Dynamic Time Warping (DTW) [25], Time Delay Embedding (TDE), ScanMatch [14], Recursive Metric (REC) [3] and Sequence Score(SS) [55]. LEV [41] is a metric that quantifies the dissimilarity between two strings by measuring the minimum number of single-character edit operations needed to transform one string into another. ScanMatch [14] involves converting eye-tracking scanpaths into string sequences and then employing an algorithm similar to the Levenshtein distance to compare the similarity of the two sequences. DTW [25] and TDE [52] are metrics based on time series analysis, while REC [3] is a metric for recursive quantification analysis. SS [55] is calculated by converting scanpaths into strings of fixation cluster IDs, and then measuring the similarity between two strings using a string matching algorithm [37].

For each image, every model generates $m = 10$ predicted scanpaths \widehat{X} , which are evaluated against all real human scanpaths X , and then averaged to determine the final prediction score. Note that our model predicts scanpath with a length of 30. When comparing across different datasets, we truncate the predicted scanpath to the length of the actual scanpath before conducting the comparison. Taking DTW [25] as an example, the score can be calculated as follows:

$$DTW = \frac{1}{n} \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m DTW(\widehat{X}_j, X_i) \quad (15)$$

where \widehat{X}_j is the j -th predicted scanpath, X_i is the i -th real human scanpath, n is the number of the real human scanpaths, m is the number of predicted scanpaths.

4.3 Implementation Details

The internal vector dimension of 3D Transformer Encoder is set to $D = 128$. We utilize a multi-head attention mechanism with 8 attention heads. The hidden layer size of the feed-forward network (FFN) is set to 64. In the 3D mixture density network model, we configure $K = 5$ Gaussian kernels, and the internal hidden layer dimension is set to 16. During the training process, our batch size is set to 18. As for the optimizer, we use AdamW [26] with a learning rate of $1e-5$. We employ a learning rate warm-up and stage-wise adjustment strategy. The warm-up period lasts for 10 epochs, and the learning rate is halved every 10 epochs. The total training duration is 50 epochs.

Our experiments were conducted on a single NVIDIA GeForce RTX 3090, and the development was conducted using the PyTorch framework.

4.4 Performance Comparison

We compared our model with SaltiNet [6], ScanGAN [35], and ScanDMM [45], utilizing the models provided officially by their respective sources to obtain prediction results. These three models are all designed for scanpath prediction on 360° images. PathGAN [5] and Zhu *et al.* [58] are two additional models specifically designed for generating scanpath on 360° images. However, it has been demonstrated that these models are unable to effectively generate realistic scanpaths [35]. Therefore, we have not included them in our comparison.

We additionally established two baseline reference values to assess the relative quality of the generated scanpath. The first one is the random baseline (Random walk). This baseline randomly generate ten scanpaths for each image, which provides the reference for lowest performance. The second one is the human baseline (Human), which compares each real human scanpath in the dataset with all the other real human scanpaths and taking the average. This baseline represents the average similarity among real human scanpaths, serving as a reference for highest performance.

Quantitative evaluation. Tab. 1 shows the quantitative comparison results between our method and other competitors on the metrics of LEV [41], DTW [25], TDE [52], ScanMatch [14], REC [3] and SS [55]. As we can see, our method outperforms all the competitors on all four datasets, achieving performance close to ‘Human’.

Table 1: Performance of scanpath prediction models on different datasets. The best result for each metric is highlighted in bold, while the second-best result is underlined.

Method	Sitzmann						Salient360!					
	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑
Random walk	51.522	2372.387	28.038	0.395	1.514	0.198	43.670	2062.080	29.753	0.375	1.381	0.176
SaltiNet [6]	51.151	2246.057	26.303	0.374	1.734	0.198	41.042	1860.397	25.001	0.429	2.250	0.220
ScanGAN [35]	45.852	1950.488	19.060	0.477	2.960	0.265	38.977	1752.371	<u>19.527</u>	0.460	3.041	0.244
ScanDMM [45]	<u>45.748</u>	<u>1939.980</u>	<u>19.098</u>	<u>0.485</u>	<u>3.156</u>	<u>0.270</u>	<u>38.545</u>	<u>1706.661</u>	20.238	<u>0.467</u>	<u>3.125</u>	<u>0.257</u>
Ours	44.691	1939.662	19.887	0.486	4.154	0.280	36.277	1515.883	19.261	0.484	4.250	0.302
Human	42.006	1841.534	16.192	0.526	5.163	0.366	35.090	1376.439	16.650	0.513	5.066	0.377
Method	AOI						JUFE					
	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑
Random walk	15.866	666.652	38.164	0.293	5.650	0.241	26.439	1270.893	38.164	0.352	1.847	0.191
SaltiNet [6]	14.978	622.079	35.724	0.337	6.306	0.245	26.115	1264.492	32.261	0.364	1.972	0.192
ScanGAN [35]	14.473	569.549	32.602	0.361	<u>7.379</u>	0.258	24.259	1110.826	25.937	0.431	3.208	0.217
ScanDMM [45]	<u>14.185</u>	569.170	32.558	<u>0.371</u>	7.328	<u>0.249</u>	<u>23.372</u>	<u>1081.465</u>	25.787	<u>0.452</u>	<u>3.527</u>	0.221
Ours	13.904	561.895	30.987	0.376	8.286	<u>0.225</u>	22.595	1054.223	22.955	0.462	4.449	<u>0.218</u>
Human	14.317	501.952	30.248	0.367	13.008	0.377	18.445	1030.970	20.305	0.564	7.864	0.330

Qualitative evaluation. We visualized the prediction results of all methods on one image of each dataset, as shown in Fig. 2. Similarly, we visualized the ground truth of human scanpath accordingly. The transition from purple to red in the scanpath indicates the beginning and end of the scanpath, respectively. It is noticeable that most ground truth scanpaths are located within the salient regions near the equator. SaltiNet [6], although designed for 360° images, suffers from large displacements. In comparison, the methods proposed by ScanGAN [35] and ScanDMM [45] have the capability to generate relatively realistic scanpaths to some extent. However, our model, in contrast to these approaches, excels at capturing salient information within 360° images and is capable of generating scanpath that are closer to the real human scanpaths. In addition, a notable flaw in the tower image is that our model tends to focus more on the prominent plants at both ends rather than the central tower, which is a common issue shared by other models as well.

4.5 Saliency Comparison

A good scanpath prediction model often assigns more attention to salient regions during the prediction process. In our study, we generated 1000 scanpath and performed post-processing on all fixations to generate corresponding saliency

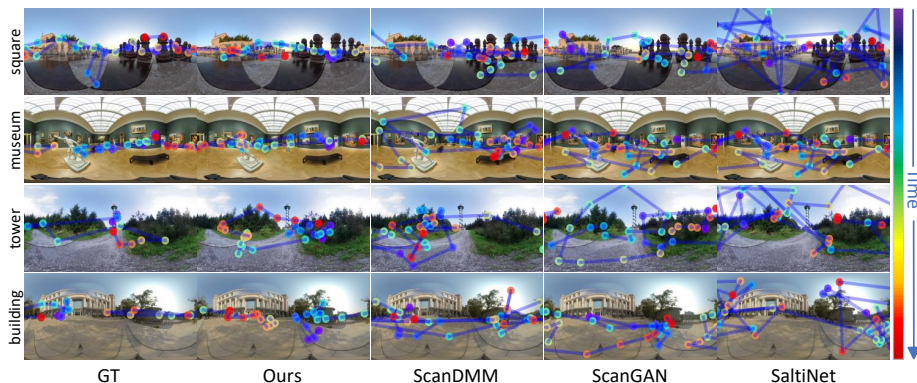


Fig. 2: Qualitative representation results of different models on four datasets. From top to bottom, the four images are sourced from Sitzmann [44], SaliNet360! [43], AOI [54], and JUFE [19], respectively. From left to right, we display the a scanpath sampled from ground truth, our model, ScanDMM [45], ScanGAN [35], and SaltiNet [6].

maps [49]. The fixation maps was obtained from all unprocessed fixations. We used sizes $H=128$ and $W=256$ for saliency maps (salmap) and fixation maps (fixmap). We compared our results with ScanGAN [35] and ScanDMM [45] on the SaliNet360! [43] and JUFE [19] datasets, under the saliency metrics AUC_Judd [9], NSS [40], CC [31], SIM [9], and KLD [29].

Table 2: Quantitative comparison results of significance comparison on SaliNet360! [43] and JUFE [19] datasets. The best result for each metric is highlighted in bold, while the second-best result is underlined.

Dataset	Method	AUC_Judd \uparrow	NSS \uparrow	CC \uparrow	SIM \uparrow	KLD \downarrow
SaliNet360! [43]	ScanGAN [35]	0.797	0.986	0.260	0.173	21.210
	ScanDMM [45]	<u>0.830</u>	1.210	<u>0.322</u>	0.183	<u>20.882</u>
	Ours	0.838	1.195	0.352	0.259	18.985
JUFE [19]	ScanGAN [35]	0.795	0.837	0.198	0.236	19.529
	ScanDMM [45]	<u>0.866</u>	1.203	0.275	<u>0.286</u>	18.248
	Ours	0.872	<u>1.035</u>	<u>0.245</u>	0.320	16.339

Based on Tab. 2, our model has demonstrated favorable performance across the majority of metrics. Our model did not achieve optimal results in a few metrics, one potential explanation is that our model exhibits a stronger emphasis on larger salient objects, which may lead to comparatively less attention towards smaller targets compared to models with more evenly distributed saliency maps. As evident in Fig. 3, ScanDMM [45] and ScanGAN [35] exhibit a characteristic of having more dispersed saliency maps. Furthermore, it can be observed that our model has the ability to focus more on salient regions compared to the other

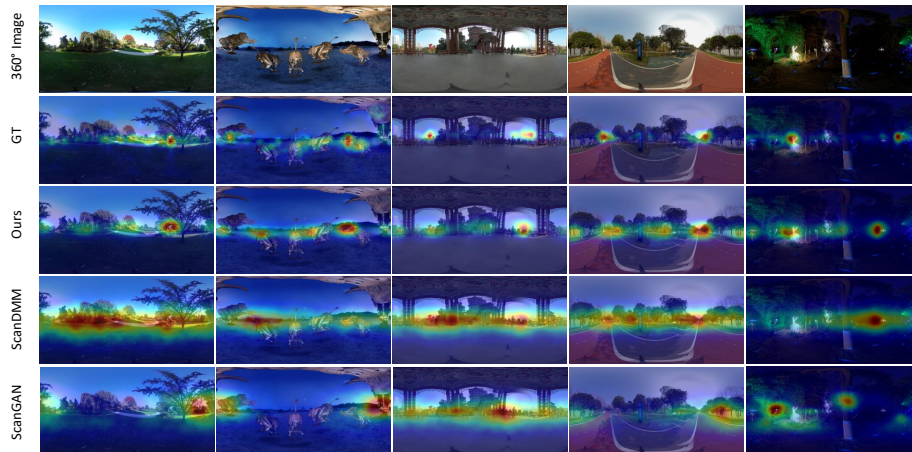


Fig. 3: Qualitative comparison results of significance comparison. The first two images are from Saliency360! [43] and the next three images are from JUFE [19]. From top to bottom are the real image, the significance maps of the real images, Ours, ScanDMM [45], ScanGAN [35].

Table 3: Efficiency comparison in terms of model size and running time

	SaltiNet [6]	ScanGAN [35]	ScanDMM [45]	Ours
Parameters(MB)	98.73	15.134	17.779	19.280
Running Time(ms)	12720	6.482	118.250	266.420

models. As a result, the post-processed saliency maps obtained from our model exhibit a closer resemblance to human-generated saliency maps.

4.6 Efficiency Comparison

We also compared our model’s size and running time for generating a scanpath with other methods. In Tab. 3, we present the model size and the runtime for generating a single scanpath (measured in milliseconds). As can be seen, we have a similar model size with other models. In terms of runtime, our model is capable of generating approximately four scanpaths of length 30 per second, which is a little slower compared to ScanGAN and ScanDMM. This is primarily because our model generates fixations in an autoregressive manner [21]. Additionally, the use of a mixture density network is another factor leading to slower running speed. However, compared to a single Gaussian distribution, multiple Gaussian distributions can more accurately represent the distribution of fixations.

4.7 Ablation Studies

We conducted ablation experiments on the Saliency360! [43] and JUFE [19] datasets. The experimental results are presented in Tab. 4, showing the optimal results

obtained for each configuration. Next, we will provide a brief overview of each model architecture. For more detailed architecture information, please refer to the supplementary file.

Table 4: Ablation studies on Salient360! [43] and JUFE [19] datasets. The best result for each metric is highlighted in bold, while the second-best result is underlined.

Method	Salient360!						JUFE					
	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑	LEV ↓	DTW ↓	TDE ↓	ScanMatch ↑	REC ↑	SS ↑
Pure ViT	37.643	1739.714	<u>19.291</u>	0.470	3.710	0.267	22.891	1072.753	23.751	0.457	3.847	0.221
Pure 2D CNN	36.848	1727.241	19.641	0.477	4.119	0.272	22.613	1060.205	<u>23.096</u>	0.463	4.181	<u>0.220</u>
Saliency	38.196	1864.914	20.059	0.475	3.156	0.253	23.833	1094.290	24.636	0.442	3.669	0.208
w/o EncoderLayer	37.352	1798.340	20.727	0.463	<u>4.171</u>	0.262	22.926	1072.263	23.196	0.455	4.260	0.218
w/o MDN + MSE Loss	39.463	1793.948	25.914	0.426	3.781	0.095	23.915	1132.127	25.675	0.416	3.833	0.102
K=1	38.643	1804.074	20.074	0.462	3.326	0.248	23.239	1084.734	23.577	0.451	3.805	0.218
K=3	36.250	2073.905	22.326	0.490	4.039	0.246	22.758	1077.365	23.302	0.460	<u>4.360</u>	0.214
K=8	38.117	1700.877	20.455	0.467	3.389	0.258	22.917	1065.240	23.470	0.455	3.914	0.210
Parallel	36.722	<u>1577.993</u>	23.726	<u>0.489</u>	3.939	<u>0.288</u>	22.567	<u>1054.644</u>	28.775	0.451	3.688	0.221
Ours	<u>36.277</u>	1515.883	19.261	0.484	4.253	0.302	<u>22.595</u>	1054.223	22.955	<u>0.462</u>	4.449	0.218

The feature extractor. To demonstrate the effectiveness of our 3D feature extractor, SphereNet, we try to employ three distinct architectures for feature extraction. Firstly, we extracted features using the patch-based approach of the Vision Transformer (ViT) method [16] (denoted as ‘Pure ViT’). Secondly, we replaced SphereNet with a standard 2D convolutional network (denoted as ‘Pure 2D CNN’). Lastly, We followed VSPT [42] to extract image features based on the saliency information. Specifically, we employed EPSNet [59] to extract saliency information (denoted as ‘Saliency’). As shown in Tab. 4, ‘Ours’ obtains better performance than ‘Pure ViT’, ‘Pure 2D CNN’, and ‘Saliency’, which demonstrates the superiority of our feature extractor.

The 3D Transformer encoder. To validate the effectiveness of our 3D Transformer Encoder, we attempted to remove the EncoderLayer in the 3D Transformer Encoder and directly utilize the feature embedding of the Transformer Encoder as input of the Transformer Decoder (denoted as ‘w/o EncoderLayer’). As we can see from Tab. 4, without our 3D Transformer Encoder, the scanpath prediction performance dropped significantly, which demonstrates the effectiveness of our 3D Transformer Encoder.

3D MDN. To validate the effectiveness of the 3D MDN, we directly exploited linear regression to predict the fixations after the Transformer Decoder. The Mean Squared Error (MSE) was utilized as the loss function (denoted as ‘w/o MDN + MSE Loss’). As can be seen in Tab. 4, ‘Ours’ outperforms ‘w/o MDN + MSE Loss’ a lot, proving the advantage of our 3D MDN.

Different number of Gaussian kernels. We compared with the model results obtained by using different numbers of Gaussian kernels in the mixture density network. These comparisons are respectively presented in the rows labeled "K=1,3,8," while our model has 5 Gaussian kernels. As we can see from Tab. 4, using 5 Gaussian kernels can provide a better representation of the fixations' distribution.

Parallel in the Transformer decoder. In our Transformer decoder, we use autoregressive method to predict one fixation each time and iteratively generate the whole fixation sequence. Through our observation, more and more models try to adopt a parallel structure in the Transformer decoder [10, 23, 36] for higher efficiency. Therefore, we also tried to build a parallel Transformer decoder in our method (denoted as 'Parallel'). In the parallel model, we use a random embedding with length L as the input to the Transformer decoder, and we can get the hidden state of a scanpath of length L at one time, all the other parts of the model are unchanged.

We can observe from Tab. 4 that the parallel structure of the decoder performs poorly than the autoregressive Transformer decoder, especially in the TDE [52] metric. Such comparison results demonstrate that an autoregressive Transformer decoder is more suitable for our scanpath prediction task.

5 Limitations and Future Works

In our work, we employed SphereNet [12] for feature extraction and 3D Mixture Density Network (MDN) for fixation generation. However, we acknowledge that these components are replaceable, and we are contemplating whether there might be better alternatives for both parts.

In addition, we are considering incorporating relevant semantic information to improve the performance further, as objects with important semantic information tend to attract human attention more easily.

6 Conclusion

Considering that the 2D equirectangular projection of 360° images has the issues of distortion and coordinate discontinuity, we proposed to execute scanpath prediction for 360° images in 3D spherical coordinate system. Correspondingly, we proposed a novel 3D Scanpath Transformer named Pathformer3D, where a 3D Transformer Encoder was first used to obtain 3D contextual feature representation of the 360° image. Then, Pathformer3D input the 3D contextual feature representation and historical fixation information into the Transformer Decoder to predict the hidden state for each fixation. Each fixation's hidden state was then fed into a 3D mixture density network to output the 3D Gaussian distribution of the fixation, from which the fixation location can be sampled. Finally, Pathformer3D model was compared with three state-of-the-art 360° image scanpath prediction models, to demonstrate the superiority of its performance.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under grant 62206127 and 62272229, the Natural Science Foundation of Jiangsu Province under grant BK20222012, and Shenzhen Science and Technology Program JCYJ20230807142001004. Rong Quan and Yantao Lai contributed equally to this work. Dong Liang is the corresponding author.

References

1. Adeli, H., Vitu, F., Zelinsky, G.J.: A model of the superior colliculus predicts fixation locations during scene viewing and visual search. *Journal of Neuroscience* **37**(6), 1453–1467 (2017)
2. Aggarwal, A., Mittal, M., Battineni, G.: Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights* **1**(1), 100004 (2021)
3. Anderson, N.C., Bischof, W.F., Laidlaw, K.E., Risko, E.F., Kingstone, A.: Recurrence quantification analysis of eye movements. *Behavior research methods* **45**, 842–856 (2013)
4. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: *ICCV*. pp. 2425–2433 (2015)
5. Assens, M., Giro-i Nieto, X., McGuinness, K., O’Connor, N.E.: Pathgan: Visual scanpath prediction with generative adversarial networks. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. pp. 0–0 (2018)
6. Assens Reina, M., Giro-i Nieto, X., McGuinness, K., O’Connor, N.E.: Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. pp. 2331–2338 (2017)
7. Boccignone, G., Ferraro, M.: Modelling gaze shift as a constrained random walk. *Physica A: Statistical Mechanics and its Applications* **331**(1-2), 207–218 (2004)
8. Brockmann, D., Geisel, T.: The ecology of gaze shifts. *Neurocomputing* **32**, 643–650 (2000)
9. Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., Torralba, A.: Mit saliency benchmark (2015)
10. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: *European conference on computer vision*. pp. 213–229. Springer (2020)
11. Clarke, A.D., Stainer, M.J., Tatler, B.W., Hunt, A.R.: The saccadic flow baseline: Accounting for image-independent biases in fixation behavior. *Journal of vision* **17**(11), 12–12 (2017)
12. Coors, B., Condurache, A.P., Geiger, A.: Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 518–533 (2018)
13. Coutrot, A., Hsiao, J.H., Chan, A.B.: Scanpath modeling and classification with hidden markov models. *Behavior research methods* **50**(1), 362–379 (2018)
14. Cristino, F., Mathôt, S., Theeuwes, J., Gilchrist, I.D.: Scanmatch: A novel method for comparing fixation sequences. *Behavior research methods* **42**, 692–700 (2010)
15. Cuturi, M., Blondel, M.: Soft-dtw: a differentiable loss function for time-series. In: *International conference on machine learning*. pp. 894–903. PMLR (2017)

16. 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 preprint arXiv:2010.11929 (2020)
17. Engbert, R., Trukenbrod, H.A., Barthelmé, S., Wichmann, F.A.: Spatial statistics and attentional dynamics in scene viewing. *Journal of vision* **15**(1), 14–14 (2015)
18. Fan, C.L., Lee, J., Lo, W.C., Huang, C.Y., Chen, K.T., Hsu, C.H.: Fixation prediction for 360 video streaming in head-mounted virtual reality. In: Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video. pp. 67–72 (2017)
19. Fang, Y., Huang, L., Yan, J., Liu, X., Liu, Y.: Perceptual quality assessment of omnidirectional images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 580–588 (2022)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems* **27** (2014)
21. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
22. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* **20**(11), 1254–1259 (1998)
23. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1780–1790 (2021)
24. Kay, A.: User interface: A personal view. *The art of human-computer interface design* pp. 191–207 (1990)
25. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowledge and information systems* **7**, 358–386 (2005)
26. Kinga, D., Adam, J.B., et al.: A method for stochastic optimization. In: International conference on learning representations (ICLR). vol. 5, p. 6. San Diego, California; (2015)
27. Krishnan, R., Shalit, U., Sontag, D.: Structured inference networks for nonlinear state space models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
28. Kümmerer, M., Bethge, M., Wallis, T.S.: Deepgaze iii: Modeling free-viewing human scanpaths with deep learning. *Journal of Vision* **22**(5), 7–7 (2022)
29. Kümmerer, M., Wallis, T.S., Bethge, M.: Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences* **112**(52), 16054–16059 (2015)
30. Le Meur, O., Coutrot, A.: Introducing context-dependent and spatially-variant viewing biases in saccadic models. *Vision research* **121**, 72–84 (2016)
31. Le Meur, O., Le Callet, P., Barba, D.: Predicting visual fixations on video based on low-level visual features. *Vision research* **47**(19), 2483–2498 (2007)
32. Le Meur, O., Liu, Z.: Saccadic model of eye movements for free-viewing condition. *Vision research* **116**, 152–164 (2015)
33. Li, C., Zhang, W., Liu, Y., Wang, Y.: Very long term field of view prediction for 360-degree video streaming. In: 2019 IEEE conference on multimedia information processing and retrieval (MIPR). pp. 297–302. IEEE (2019)

34. Liu, H., Xu, D., Huang, Q., Li, W., Xu, M., Lin, S.: Semantically-based human scanpath estimation with hmms. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3232–3239 (2013)
35. Martin, D., Serrano, A., Bergman, A.W., Wetzstein, G., Masia, B.: Scangan360: A generative model of realistic scanpaths for 360 images. *IEEE Transactions on Visualization and Computer Graphics* **28**(5), 2003–2013 (2022)
36. Mondal, S., Yang, Z., Ahn, S., Samaras, D., Zelinsky, G., Hoai, M.: Gazeformer: Scalable, effective and fast prediction of goal-directed human attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1441–1450 (2023)
37. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* **48**(3), 443–453 (1970)
38. Nguyen, A., Yan, Z., Nahrstedt, K.: Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1190–1198 (2018)
39. 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 preprint arXiv:1701.01081 (2017)
40. Peters, R.J., Iyer, A., Itti, L., Koch, C.: Components of bottom-up gaze allocation in natural images. *Vision research* **45**(18), 2397–2416 (2005)
41. Privitera, C.M., Stark, L.W.: Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on pattern analysis and machine intelligence* **22**(9), 970–982 (2000)
42. Qiu, M., Rong, Q., Liang, D., Tu, H.: Visual scanpath transformer: Guiding computers to see the world. In: 2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 223–232. IEEE (2023)
43. Rai, Y., Gutiérrez, J., Le Callet, P.: A dataset of head and eye movements for 360 degree images. In: Proceedings of the 8th ACM on Multimedia Systems Conference. pp. 205–210 (2017)
44. Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., Masia, B., Wetzstein, G.: Saliency in vr: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics* **24**(4), 1633–1642 (2018)
45. Sui, X., Fang, Y., Zhu, H., Wang, S., Wang, Z.: Scandmm: A deep markov model of scanpath prediction for 360° images. In: IEEE Conference on Computer Vision and Pattern Recognition (2023)
46. Sun, W., Chen, Z., Wu, F.: Visual scanpath prediction using ior-roi recurrent mixture density network. *IEEE transactions on pattern analysis and machine intelligence* **43**(6), 2101–2118 (2019)
47. Sun, X., Yao, H., Ji, R., Liu, X.M.: Toward statistical modeling of saccadic eye-movement and visual saliency. *IEEE Transactions on Image Processing* **23**(11), 4649–4662 (2014)
48. Tatler, B.W., Brockmole, J.R., Carpenter, R.H.: Latest: A model of saccadic decisions in space and time. *Psychological review* **124**(3), 267 (2017)
49. Upenik, E., Ebrahimi, T.: A simple method to obtain visual attention data in head mounted virtual reality. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). pp. 73–78. IEEE (2017)
50. Verma, J.K., Paul, S.: *Advances in Augmented Reality and Virtual Reality*. Springer (2022)

51. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in facebook. In: Proceedings of the 2nd ACM workshop on Online social networks. pp. 37–42 (2009)
52. Wang, W., Chen, C., Wang, Y., Jiang, T., Fang, F., Yao, Y.: Simulating human saccadic scanpaths on natural images. In: CVPR 2011. pp. 441–448. IEEE (2011)
53. Xia, C., Han, J., Qi, F., Shi, G.: Predicting human saccadic scanpaths based on iterative representation learning. *IEEE Transactions on Image Processing* **28**(7), 3502–3515 (2019)
54. Xu, M., Yang, L., Tao, X., Duan, Y., Wang, Z.: Saliency prediction on omnidirectional image with generative adversarial imitation learning. *IEEE Transactions on Image Processing* **30**, 2087–2102 (2021)
55. Yang, Z., Huang, L., Chen, Y., Wei, Z., Ahn, S., Zelinsky, G., Samaras, D., Hoai, M.: Predicting goal-directed human attention using inverse reinforcement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 193–202 (2020)
56. Zanca, D., Melacci, S., Gori, M.: Gravitational laws of focus of attention. *IEEE transactions on pattern analysis and machine intelligence* **42**(12), 2983–2995 (2019)
57. Zhu, Y., Zhai, G., Min, X.: The prediction of head and eye movement for 360 degree images. *Signal Processing: Image Communication* **69**, 15–25 (2018)
58. Zhu, Y., Zhai, G., Min, X., Zhou, J.: The prediction of saliency map for head and eye movements in 360 degree images. *IEEE Transactions on Multimedia* **22**(9), 2331–2344 (2019)
59. Zou, Z., Ye, M., Li, S., Li, X., Dufaux, F.: 360° image saliency prediction by embedding self-supervised proxy task. *IEEE Transactions on Broadcasting* (2023)