Gaze Prediction in Dynamic $360°$ Immersive Videos

Anonymous CVPR submission

Paper ID 2529

I. Transformation

Given a viewing sphere of unit radius centered at point O, as shown in Fig. 1, the FoV is modeled as a plane segment $ABCD$ tangential to the sphere at the center O$'$ of the FoV. We get O$'$ by the head information we recorded from the HMD.

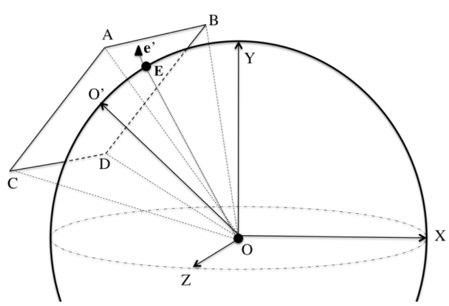

Figure 1. Example of a viewport

To determine the pixels in the FoV, we use the pinhole camera model, *i.e.* a scene view is formed by projecting 3D points onto the image plane $ABCD$ using a perspective transformation. If we uniformly span the spherical coordinates in the visible region of the sphere and pass rays from O to the points on the sphere, they will intersect the plane $ABCD$ with non-uniform spacing between the pixels. We refer to this as the **forward projection**. In order to compute a uniform grid of pixels in the FoV, we start with the desired locations in the FoV and reverse the mapping to compute corresponding locations on the sphere. We refer to this as the **backward projection**.

II. Special Saliency Detection in FoV

Given a panorama image and gaze orientation, we generate the FoV image through the forward projection mentioned in Sec. I. After that, SalNet [1] is deployed to detect the saliency region, which is regarded as candidates of gaze orientations in upcoming frames. To align the saliency maps in FOV scale with that of global scale, we transform the saliency maps in FOV scale back to panorama with backward projection. The whole pipeline is depicted in Fig. 2.

III. Optical Flow Estimation in FoV

Unlike special saliency detection in FoV, we employ another strategy to estimate optical flow in FoV. It is worth noting that for points in unit sphere, their corresponding points in plane $ABCD$ is non-uniform spread, so the pixels around the edge are stretched, which causes the optical flow in edge part is usually larger than that on object. As a result, the estimated optical flow with FOV image is incorrect compared with the real scene. Therefore, we directly estimate the optical flow in panorama image, then we do the element-wise product between the optical flow in panorama and FoV mask, and use it as an optical flow estimation in FOV scale, as shown in Fig. 3.

IV. Intersection Angle Error

For a given gaze point (x, y), where x is latitude and y is longitude, its coordinate in the unit sphere is $\mathbb{P} = (\cos x \cos y, \cos x \sin y, \sin x)$, then for a ground truth gaze point (x, y) and its predicted gaze point $(\hat{x}, \hat{y})$, we can get corresponding coordinates in unit sphere as $\mathbb{P}$ and $\hat{\mathbb{P}}$, the intersection angle error between them can be computed as

$$d = \arccos\left(<\mathbb{P},\, \hat{\mathbb{P}}>\right)$$

where $<, >$ is inner product.

V. Examples of Our Dataset

Fig. 4 show some typical $360°$ videos. Our dataset includes natural scenes and wild animals, underwater scenes, driving a plane, water activities, extreme sports, ball sports, music and dance, concert and shows. Some videos are captured with a fixed camera view, while some are shotted with a moving camera that would probably introduce more variance in eye fixation across different users.

VI. Examples of Scan Path

Fig. 5 shows more examples of viewers' scan paths during the first 40 frames. As shown in these scan paths, the

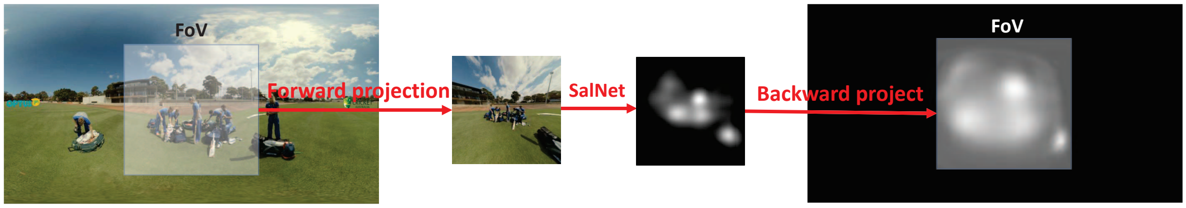

Figure 2. The pipeline of saliency detection in FoV

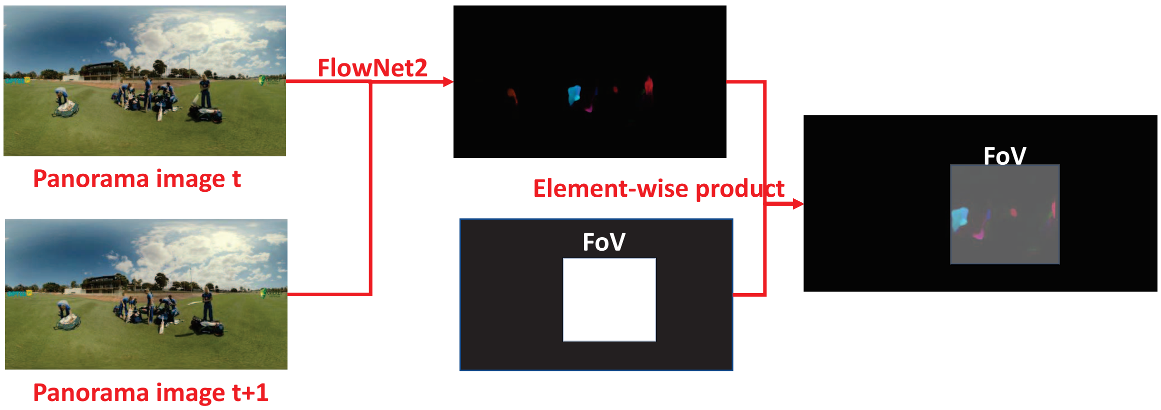

Figure 3. The pipeline of optical flow estimation in FoV

viewers follow some pattern to explore the scene in VR rather than randomly watching. So we can infer a user's gaze points in future frames based on its history scan path.

References

[1] J. Pan, E. Sayrol, X. Giroinieto, K. Mcguinness, and N. E. O-connor. Shallow and deep convolutional networks for saliency prediction. pages 598–606, 2016. 1

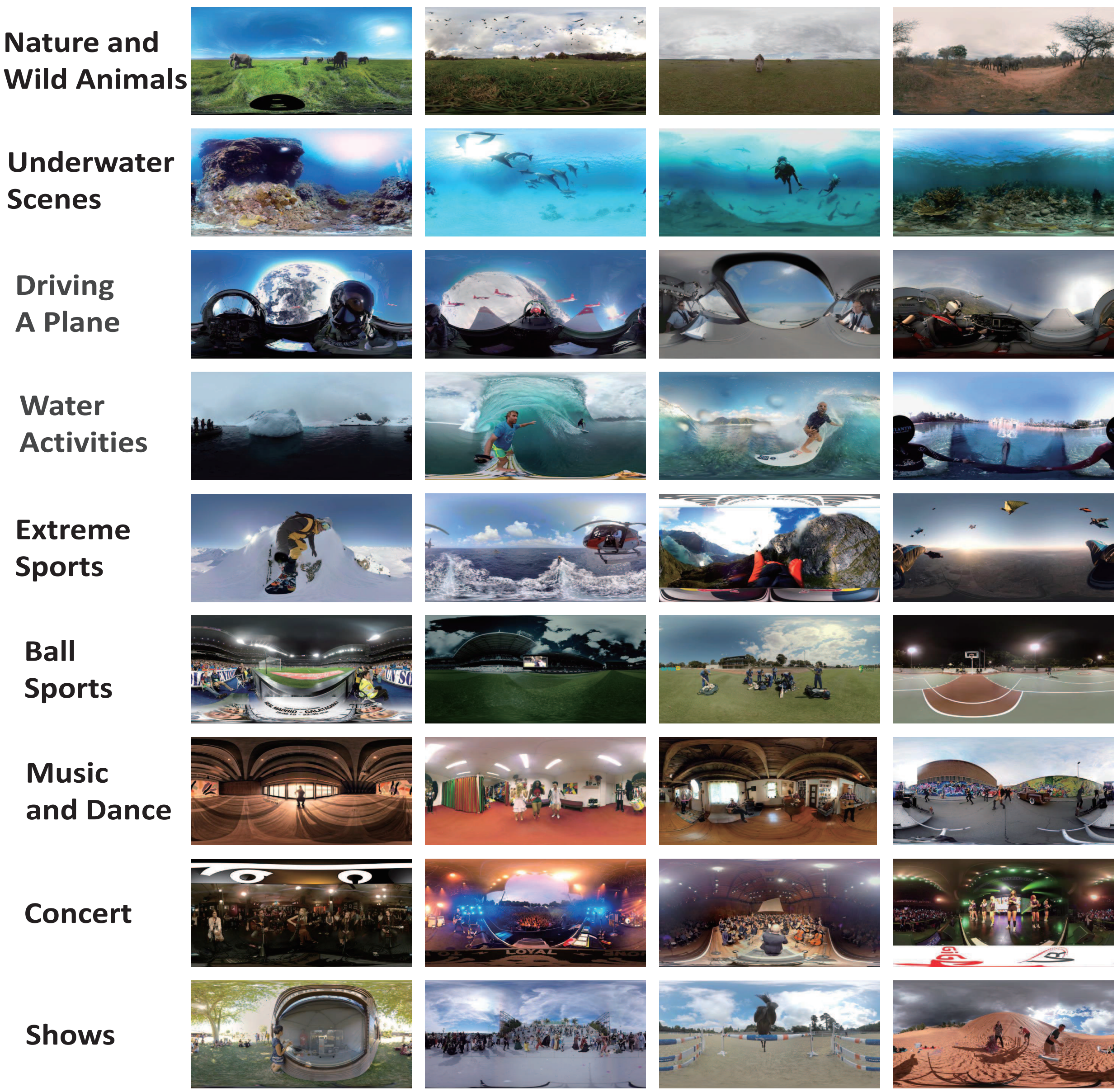

Figure 4. The examples of our Dataset

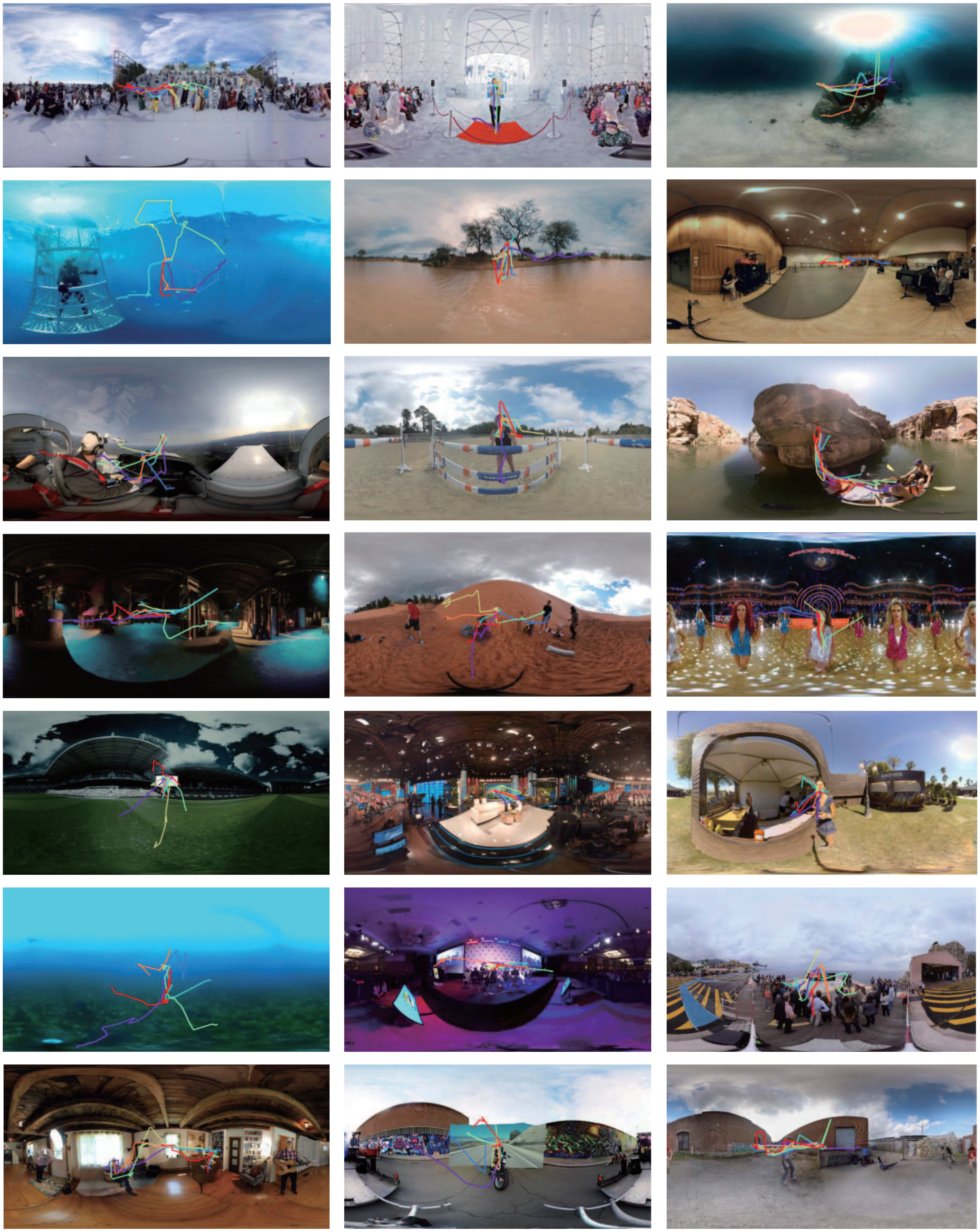

Figure 5. Some examples of viewers' scan path