Looking here or there? Gaze Following in 360-Degree Images (Supplementary Material)

Yunhao Li[1] Wei Shen[2*] Zhongpai Gao[2] Yucheng Zhu[1] Guangtao Zhai[1*] Guodong Guo[3]
[1]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University [3]Baidu
{lyhsjtu, wei.shen, gaozhongpai, zyc420, zhaiguangtao}@sjtu.edu.cn; guoguodong01@baidu.com

In our supplementary material, we first provide a detailed derivation to show how to compute the approximation of the projected 2D gaze direction $\hat{\mathbf{d}}_p^I \approx (-\hat{d}_p^S|_x, -\hat{d}_p^S|_y)$ in the local pathway of our framework (Sec. 4.4.2), where $\hat{\mathbf{d}}_p^S = (\hat{d}_p^S|_x, \hat{d}_p^S|_y, \hat{d}_p^S|_z)$ is the estimated 3D gaze direction of a human subject. We then show the gaze targets distribution in our GazeFollow360 dataset and more qualitative results of our methods.

1. Approximation of the Projected 2D Gaze Direction $\hat{\mathbf{d}}_p^I$

Given a 2D image as shown in Fig. 1, line segment $\overline{P^I Q^I}$ represents the projected 2D gaze sight line, where points P^I and Q^I are the locations of the human subject and the gaze target, respectively. Its direction $\hat{\mathbf{d}}_p^I$ can be represented by $\hat{\mathbf{d}}_p^I = (-l_1, -l_2)$ in terms of the 2D image plane coordinates, where l_1 and l_2 are the lengths of the projections of line segments $\overline{P^I Q^I}$ on X_I axis and Y_I axis, respectively. l is the length of line segment $\overline{P^I Q^I}$. To show that $\hat{\mathbf{d}}_p^I = (-l_1, -l_2) \approx (-\hat{d}_p^S|_x, -\hat{d}_p^S|_y)$, we need to show that $\frac{l_2}{l_1} \approx \frac{\hat{d}_p^S|_y}{\hat{d}_p^S|_x}$.

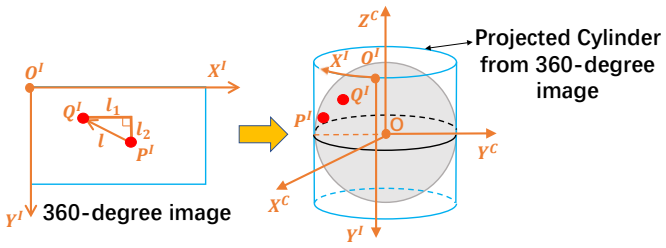

Figure 1. Image-to-sphere coordinate transformation. The 360-degree image plane is projected to the surface of the cylinder.

Towards this end, we separately calculate the relation between $\hat{\mathbf{d}}_p^I$ and $\hat{\mathbf{d}}_p^S$ from different views of image-to-sphere projection. Let us first review the process of image-to-sphere projection to help understand our calculation. As shown in Fig. 1, a 360-degree image is first rolled up to a surface of a cylinder, then the cylinder surface is wrapped to a sphere. Thus, the cylinder is shown as the blue circle from the top view (Fig. 2(a)) and the blue square from the side view (Fig. 2(b)). Now, we provide the detail of our calculation. From the top view, we find that the curve length of $\overset{\frown}{P^I Q^I}$ is exactly l_1, and curve $\overset{\frown}{PQ}$ represents the projected gaze sight line on the sphere from the top view. Note that P and Q are closed to each other on the sphere for a short-distance gaze behavior which indicates that $\overset{\frown}{PQ}$ can be approximated by $\overline{PQ'}$. Thus the curve length l_x^S of $\overset{\frown}{PQ}$ is:

$$l_x^S \approx l_1 \cdot \cos\varphi_p, \tag{1}$$

where φ_p represents the latitude of point P on the sphere. Similarly, from the side view, as shown in Fig. 2 (b), the curve length l_y^S of $\overset{\frown}{PQ}$ can be approximated by:

$$l_y^S \approx \frac{l_2}{\cos\varphi_p}. \tag{2}$$

Here, the length of line segment $\overline{P^I Q^I}$ along the border of the square is exactly l_2. Thus, by combining Eq. 1 and Eq. 2, we have

$$\frac{l_2}{l_1} = \frac{l_y^S \cdot \cos\varphi_p}{\frac{l_x^S}{\cos\varphi_p}} = \frac{l_y^S}{l_x^S} \cdot \cos\varphi_p^2. \tag{3}$$

Note that $l_x^S = r\alpha\cos\varphi_p$ and $l_y^S = r\theta$, where r is the radius of the sphere, α is the longitude difference between point P and Q and θ is the latitude difference between P and Q. Then, we have

$$\frac{l_2}{l_1} = \frac{l_y^S}{l_x^S}\cos\varphi_p^2 = \frac{r\theta}{r\alpha\cos\varphi_p} = \frac{\theta}{\alpha}\cos\varphi_p \tag{4}$$

Additionally, when capturing 360-degree images, photographers tend to place 360-degree cameras at similar heights

*Corresponding author: Guangtao Zhai, Wei Shen

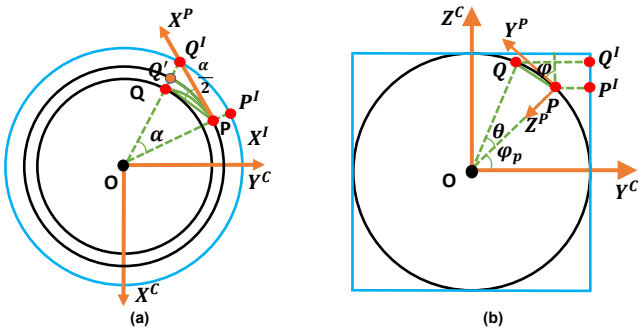

Figure 2. From the top view (a) and the side view (a) to observe how the the cylinder surface (in blue) is wrapped to the sphere (in black). Point P^I and Point Q^I are located on the surface of cylinder. Point P and Point Q are located on the sphere. α and θ are the latitude difference and longitude difference between point P and Q, respectively. Q' is a point whose latitude and longitude equal to the latitude of P and the longitude of Q, respectively. φ_p is the latitude of point P on the sphere.

as human subjects, which means φ_p is usually small, This indicates that

$$\frac{l_2}{l_1} \approx \frac{\theta}{\alpha}. \tag{5}$$

Now we calculate $\frac{\hat{d}_p^S|_y}{\hat{d}_p^S|_x}$. As shown in Fig. 2(a), we find that $\hat{d}_p^S|_x$ is the projection of line segment $\overline{PQ}$ on the X^P axis, from the top view. Since P and Q are close to each other, curve $\widehat{PQ}$ can be approximated by line $\overline{PQ}$, thus we have

$$\hat{d}_p^S|_x \approx r\alpha \cos\frac{\alpha}{2}. \tag{6}$$

Again, since P and Q are near to each other, the longitude difference α between them is small, then $\cos\frac{\alpha}{2} \approx 1$. So, Eq. 6 becomes

$$\hat{d}_p^S|_x \approx r\alpha. \tag{7}$$

As shown in Fig. 2(b), $\hat{d}_p^S|_y$ is the projection of line $\overline{PQ}$ on the Y^P axis from the side view. Similarly, we have

$$\hat{d}_p^S|_y \approx r\theta \cos\frac{\theta}{2} \approx r\theta \tag{8}$$

Then, we obtain

$$\frac{\hat{d}_p^S|_y}{\hat{d}_p^S|_x} = \frac{\theta}{\alpha}. \tag{9}$$

Finally, by combining Eq. 1 and Eq. 1, we obtain

$$\hat{\mathbf{d}}_p^I \approx (-\hat{d}_p^S|_x, -\hat{d}_p^S|_y). \tag{10}$$

2. Implementation Details

We implement our whole framework on PyTorch. The three modules in our framework are trained in a stage-wise manner. For convenience, we first train the GDE module, then we fix the GDE module and jointly train the DP and DF modules together. The whole network is optimized by Adam[2], with learning rate of 0.0001.

3. Gaze Target Distribution in GazeFollow360 dataset

In this section, we provide the histogram of circles of gaze targets in our GazeFollow360 dataset, which is shown in Fig. 3. Our dataset has a large variation of gaze targets including faces, human bodies and diverse man-made objects.

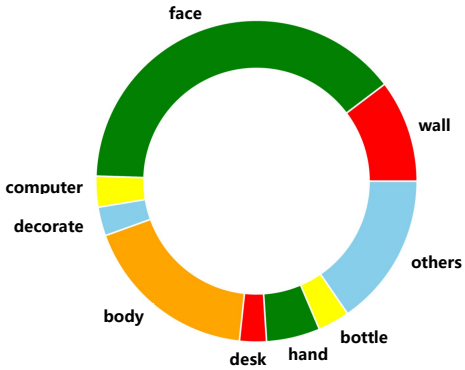

Figure 3. Histogram over the categories of the gaze targets objects in GazeFollow360 dataset.

4. Qualitative Results

In this section, we provide more qualitative results on the GazeFollow360 dataset, which are shown in Fig. 4. We observe that our methods performs much better than two 2D gaze following methods and one 3D gaze estimation method(Lian *et al.* [1], Chong *et al.* [3],Zhang *et al.* [4]) and can cope with both short-distance and long-distance gaze behaviors.

References

[1] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M. Rehg. Detecting attended visual targets in video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[3] Dongze Lian, Zehao Yu, and Shenghua Gao. Believe it or not, we know what you are looking at! In *Asian Conference on Computer Vision*, pages 35–50. Springer, 2018. 2, 3

[4] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–60, 2017. 2, 3

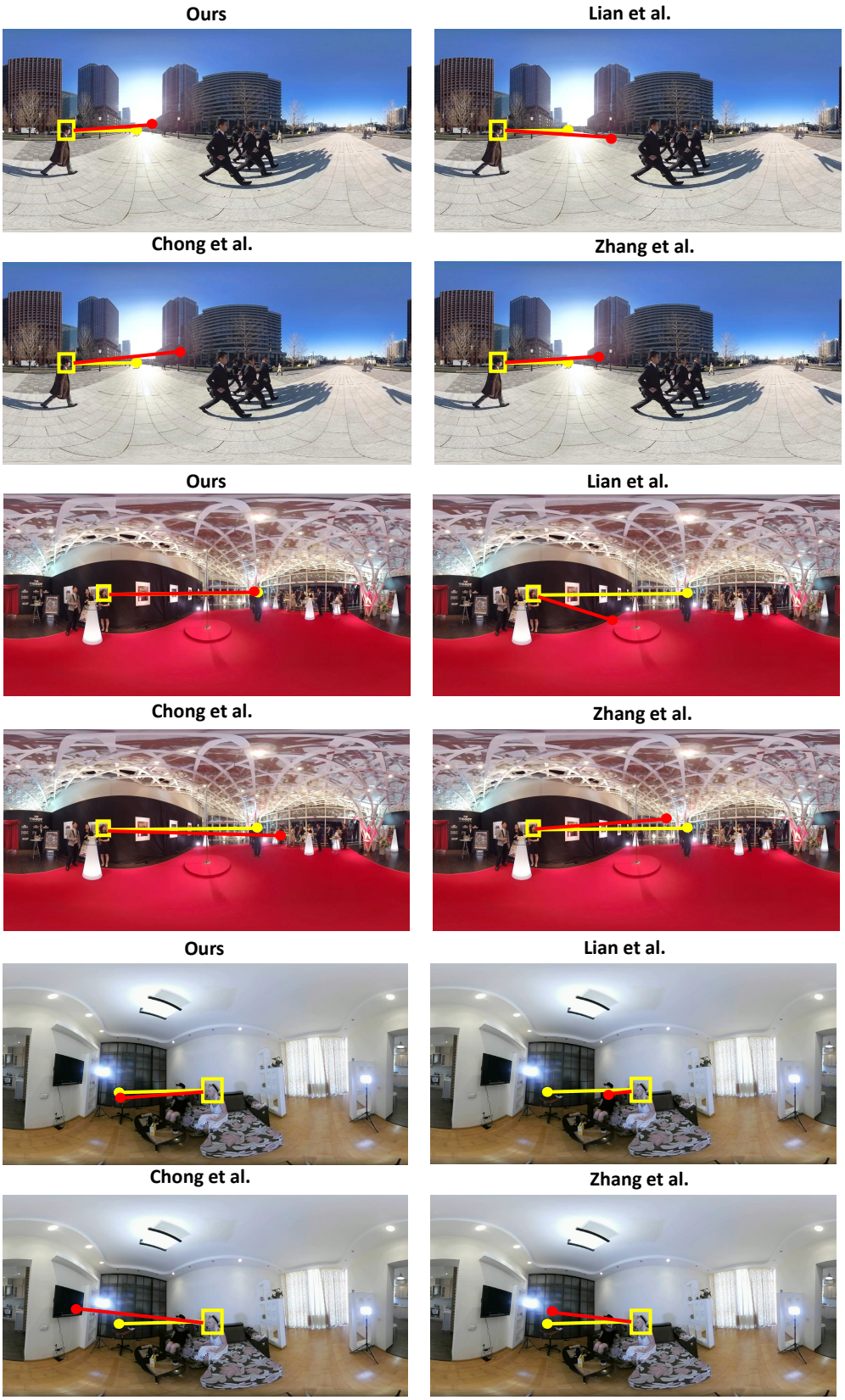

Figure 4. Qualitative results on the GazeFollow360 dataset. In each example, we showthe results of our method, Lian *et al.* [3], Chong *et al.* [1] and Zhang *et al.* [4]. The yellow and red points are ground truth and predicted targets, respectively.