What Do You See in Vehicle? Comprehensive Vision Solution for In-Vehicle Gaze Estimation

Supplementary Material

Due to the page limitation, we present some details in the supplementary material. We first describe the details of method and then demonstrate more experiment results.

1. Methodology

1.1. Gaze Target Calibration

We uses a transparent chessboard for gaze target calibration. The mathematical deduction is shown in this section.

We set a transparent chessboard between the DMS camera and the depth camera. The two cameras both capture one side of the chessboard. Therefore, we can compute the pose matrices of the two cameras w.r.t. the chessboard coordinate system. We denote the pose matrices of the two cameras as $\{\mathbf{R}_{\mathrm{dms}}, \mathbf{t}_{\mathrm{dms}}\}$ and $\{\mathbf{R}_{\mathrm{depth}}, \mathbf{t}_{\mathrm{depth}}\}$. Given a point $\mathbf{p}_1$ in the chessboard coordinate system, we have

$$\mathbf{p}_{\mathrm{dms}} = \mathbf{R}_{\mathrm{dms}}\mathbf{p}_1 + \mathbf{t}_{\mathrm{dms}}. \tag{1}$$

Similarly, we can compute the 3D position $\mathbf{p}_{\mathrm{depth}}$ in the depth camera coordinate system with a given point $\mathbf{p}_2$.

$$\mathbf{p}_{\mathrm{depth}} = \mathbf{R}_{\mathrm{depth}}\mathbf{p}_2 + \mathbf{t}_{\mathrm{depth}}. \tag{2}$$

Note that, $\mathbf{p}_1$ and $\mathbf{p}_2$ represent points in two different chessboard coordinate systems since the two cameras respectively capture each side of the chessboard. We further derive the rotation matrix and the translation matrix between the two chessboard coordinate systems. We use $\{\mathbf{R}_{\mathrm{chess}}, \mathbf{t}_{\mathrm{chess}}\}$ to represent them and have

$$\mathbf{p}_1 = \mathbf{R}_{\mathrm{chess}}\mathbf{p}_2 + \mathbf{t}_{\mathrm{chess}} \tag{3}$$

We have $\mathbf{R}_{\mathrm{chess}} = \mathrm{diag}\,(0, 0, -1)$ and $\mathbf{t}_{\mathrm{chess}} = (0, 0, -\mathrm{d})$, where d is the thickness of the chessboard. Note that, some cameras will capture images in a mirror mode. The rotation matrix should be adjusted based on real setting.

Therefore, given a point $\mathbf{p}_{\mathrm{depth}}$ in the depth camera coordinate system, we can obtain the $\mathbf{p}_{\mathrm{dms}}$ using Eq. (1), Eq. (2) and Eq. (3). We use $\mathbf{R}_{\mathrm{rot}}$ and $\mathbf{t}_{\mathrm{rot}}$ to represent the rotation and translation matrices between the depth and DMS cameras. It is easy to derive that

$$\mathbf{R}_{\mathrm{rot}} = \mathbf{R}_{\mathrm{dms}}\mathbf{R}_{\mathrm{chess}}\mathbf{R}^{-1}_{\mathrm{depth}}, \tag{4}$$

and

$$\mathbf{t}_{\mathrm{rot}} = -\mathbf{R}_{\mathrm{dms}}\mathbf{R}_{\mathrm{chess}}\mathbf{R}^{-1}_{\mathrm{depth}}\mathbf{t}_{\mathrm{depth}} + \mathbf{R}_{\mathrm{dms}}\mathbf{t}_{\mathrm{chess}} + \mathbf{t}_{\mathrm{dms}}. \tag{5}$$

We can use following equation for the conversion.

$$\mathbf{p}_{\mathrm{dms}} = \mathbf{R}_{\mathrm{rot}}\mathbf{p}_{\mathrm{depth}} + \mathbf{t}_{\mathrm{dms}}. \tag{6}$$

1.2. Implementation details of GazeDPTR

In this paper, we propose a GazeDPTR for gaze estimation. We also extend the GazeDPTR for gaze zone classification. We train the extended network in an end-to-end manner.

In detail, GazeDPTR contains two GazePTRs for feature extraction from original and normalized images. GazePTR is modified based on GazeTR [1]. We use ResNet18 to extract multi-level feature maps and obtain 4 different scale feature maps. Their scales are $64 \times 56 \times 56$, $128 \times 28 \times 28$, $256 \times 14 \times 14$, $512 \times 7 \times 7$. We use 1×1 convolution layers and global average pooling layers to convert them into 128D features. We denote these features as $\{f_i \in \mathbb{R}^{128}\}_{i=1,2,3,4}$. We use sup ^{n} and ^{o} to represent the feature is extracted from normalized or original images, *e.g.* f^o_1. Next, we use a 6-layer transformer to integrate these features for the final feature. We use one learnable token to aggregate $\{f^n_i\}$ for f^n_{final}. Two learnable tokens are used to aggregate $\{f^o_i\}$ since we need feature f^o_{final} for gaze estimation and visual feature f_{visual} for gaze zone classification. We use another 6-layer transformer to integrate f^n_{final} and f^o_{final} for f_{gaze}. We add a MLP to estimate gaze directions from f_{gaze}.

We project the estimated gaze into a tri-plane. Note that, we cut off the propagation of gradient in this operation layer since it drops gaze estimation accuracy but cannot improve gaze classification performance. We use a 2-layer transformer to extract positional feature f_{pos} where a deep transformer will vanish gradients. We also use a 6-layer transformer to integrate positional features and visual features for f_{zone}. We add a MLP to predict gaze zone from f_{zone}.

Regarding the loss function, we use L1 loss $\mathcal{L}_{gaze}$ for the gaze estimation task. Our method contains two ground truths $\mathbf{g}^o$ and $\mathbf{g}^n$. We define the function $\mathcal{L}^o_{gaze}(f)$ that means we set a MLP to estimate gaze from feature f and measure the L1 distance between the gaze and $\mathbf{g}^o$ for loss function. The same for $\mathcal{L}^n_{gaze}(f)$.

we require following feature should be gaze-related including 1) multi-level feature $\{f^n_i\}$ 2) intermediate features f^n_{final} and f^o_{final} 3) gaze feature f_{gaze}. The loss function can be represented as:

$$\mathcal{L}_1 = \sum_{i=1}^{4}\sum_{j\in\{o,n\}}\mathcal{L}^j_{gaze}(f^j_i) + \sum_{j\in\{o,n\}}\mathcal{L}^j_{gaze}(f^j_{final}) + \mathcal{L}^n_{gaze}(f_{gaze}) \tag{7}$$

We set cross entropy loss as the loss function for gaze zone classification. We also define the loss $\mathcal{L}_{zone}(f)$ that means we set a MLP to predict gaze zone from f and mea-

Table 1. We define nine zones for gaze zone classification and show the average precision (%) on each zone.

Visual Feature	Positional Feature	Left-side mirror	Rear-view mirror	Right-side mirror	Central-control screen	Steering wheel	Handbrake	Dashboard	Left-side windshield	Right-side windshield	Avg
✓		96.5	47.0	68.4	83.4	92.5	79.9	72.5	89.3	69.9	79.4
	✓	97.3	39.1	79.5	63.7	87.6	54.4	45.2	88.9	65.6	75.3
✓	✓	96.9	42.1	75.7	77.1	92.0	86.9	77.5	89.7	75.4	81.8

Table 2. We define one additional class *None* to account for samples that do not fall within the nine zones. We respectively report the average precision *with* and *w/o* the *None* region.

Visual Feature	Positional Feature	AP *w/o None* region	AP *with None* region
✓		79.4%	78.1%
	✓	75.3%	75.8%
✓	✓	81.8%	80.0%

Table 3. We selected best results from a pair of original and normalized images. The performance shows the selected result significantly outperform each of images.

	Original images	Normalized images	Selected
Acc	7.44°	7.04°	5.72°

sure the cross entropy loss. The loss function for gaze zone task is

$$\mathcal{L}_2 = \mathcal{L}_{zone}(f_{pos}) + \mathcal{L}_{zone}(f_{visual}) + \mathcal{L}_{zone}(f_{zone}) \quad (8)$$

We optimize the whole network using

$$\mathcal{L}_{GazeDPTR} = \mathcal{L}_1 + \mathcal{L}_2 \quad (9)$$

2. Additional Experiments

2.1. Setup of Gaze Zone Classification

Our paper extends gaze estimation for gaze zone classification. In this section, we provide details about the experimental setup.

During data collection, stickers are placed strategically within the vehicle. Based on the positions of these stickers, we divide the in-vehicle region into nine zones: left-side mirror, right-side mirror, rear-view mirror, steering wheel, left-side windshield, right-side windshield, central-control screen, handbrake, and dashboard. It is important to note that the dashboard encompasses not only the instrument cluster behind the steering wheel but also the air conditioning panel. Additionally, we introduce an extra region *None* to account for points that do not fall within the specified nine zones. The performance of this additional class is not included in the average performance calculation.

The average precision (AP) of each classes is shown in the Fig. 1. GazeDPTR integrates two features and show better average AP. We also show the average performance with *None* region in Tab. 2 for reference.

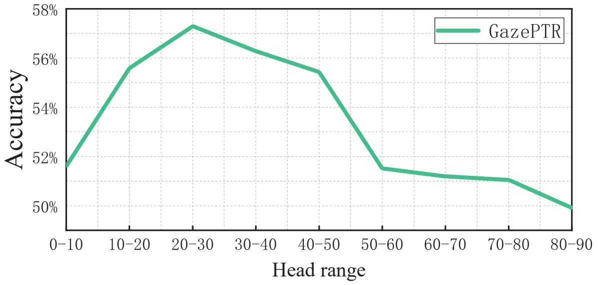

Figure 1. We count the improvement ratio in each head range. A larger ratio means more samples have performance improvement due to normalization. The result demonstrates that the large head range usually has relatively low improvement ratio.

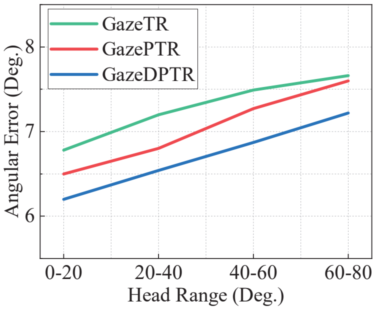

Figure 2. We count the average angular error in different head ranges. GazePTR estimates gaze from normalized images while GazeDPTR uses both normalized and original images for gaze estimation. It is interesting that GazeDPTR has larger performance improvement in a large head range than GazePTR. Combining with the result in Fig. 1, the reason may be the relatively low improvement ratio in the large head range.

2.2. Analysis on Normalized and Original Images

Our work uses both original and normalized images for gaze estimation. Our hypothesis is that the conbination of two images can provide additional insights beyond what each offers. In this section, we show experimental result to validate our hypothesis. We initially conducted an oracle baseline by separately training GazePTR on both the original and normalized datasets and selecting the best result from each image pair. The result is shown in Tab. 3. The selected performance demonstrated a remarkable improvement, achieving 5.72°, which significantly surpasses the performances in both the original and normalized images.

To gain a more nuanced understanding of the improvement across different head pose ranges, we calculated the

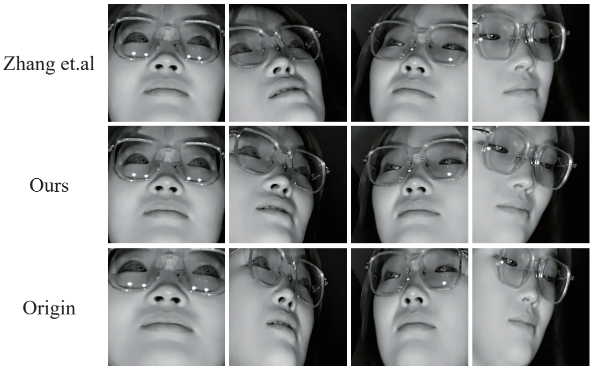

Figure 3. We show the normalization image obtained from Zhang*et al*. [2] and ours. We also visual the original images which are directly cropped from scene images. Zhang *et al*. rotate images based on the x-axis of head. It sometimes produce unstable result in extreme head pose, *e.g*., the second column. We modify their method and cancel such rotation. Our method has better performance which is shown in our manuscript.

improvement ratio. A particular sample is considered improved if the performance of the normalized image surpasses that of the original image. The results are visualized in Fig. 1, where images that failed in the large head pose range typically exhibit a relatively low improvement ratio. Additionally, the angular error across different head poses is depicted in Fig. 2, underscoring the larger performance improvement in a significant head pose range for GazeDPTR. These findings provide valuable insights into the advantages of our proposed method.

2.3. Visualization of Normalization Images

We show the images of different normalization methods and original images in Fig. 3. Zhang *et al*. [2] rotate images based on the x-axis of head. It sometimes produce unstable result in extreme head pose, *e.g*., the second column in Fig. 3. We modify their method and cancel such rotation. Our method has better performance which is shown in our manuscript.

References

[1] Yihua Cheng and Feng Lu. Gaze estimation using transformer. *ICPR*, 2022. 1

[2] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications*, 2018. 3