Supplementary Materials: ESCNet: Gaze Target Detection with the Understanding of 3D Scenes

Jun Bao[1] Buyu Liu[2] Jun Yu[1*]
[1]Hangzhou Dianzi University [2]NEC Laboratories America

In this supplementary material, we include further details on the following:

- Detailed steps to obtain hyper-parameters a, b and focal length c.
- Details of model structure and training procedure.
- More analysis on our method.

Our code, model and generated data, including 3D point clouds and 3D ground truth, will be publicly available.

1. Hyper-parameter a, b and focal length c

We first remind the reviewers of the definitions of our variables in Fig. 1. Then we provide more details about how to compute a, b and c with these variables.

For each person $l = [1, \ldots, N_i]$ in image I_i, we denote its head size as h_i^l and its average absolute depth as $d_i^{r,h,l*}$. Then we have:

$$d_i^{r,h,l*} = e_i^{h,l} \cdot h_i^l \cdot c = \frac{1}{a \cdot (d_i^{h,l} + b)}$$

Then we get the equation $a \cdot c \cdot h_i^l = \frac{1}{e_i^{h,l} \cdot (d_i^{h,l} + b)}$. Assuming that h_i^l are of the same size for all l, our goal becomes to minimize the variance of $\frac{1}{e_i^{h,l} \cdot (d_i^{h,l} + b)}$ w.r.t. b for all l in image I_i:

$$minimize \quad \textbf{Var}(e_i^{h,l} \cdot (d_i^{h,l} + b)) \Rightarrow$$

$$\frac{1}{N_i - 1} \sum_{l=1}^{N_i} [e_i^{h,l} d_i^{h,l} + e_i^{h,l} b - \frac{\sum_{m=1}^{N_i} (e_i^{h,m} d_i^{h,m} + e_i^{h,m} b)}{N_i}]^2$$

$$\Rightarrow \sum_{l=1}^{N_i} [e_i^{h,l} d_i^{h,l} - \frac{\sum_{m=1}^{N_i} e_i^{h,m} d_i^{h,m}}{N_i} + (e_i^{h,l} - \frac{\sum_{m=1}^{N_i} e_i^{h,m}}{N_i}) b]^2$$

$$\Rightarrow \sum_{l=1}^{N_i} (e_i^{h,l} - \overline{e_i^{h,l}})^2 b^2 + 2 \sum_{l=1}^{N_i} (e_i^{h,l} - \overline{e_i^{h,l}})(e_i^{h,l} d_i^{h,l} - \overline{e_i^{h,l} d_i^{h,l}}) b$$

where $\overline{*} = \frac{\sum_l (*)}{N_i}$ denotes the average function over $*$. Then we differentiate the above mentioned equation and we have:

$$2b \sum_{l=1}^{N_i} (e_i^{h,l} - \overline{e_i^{h,l}})^2 + 2 \sum_{l=1}^{N_i} (e_i^{h,l} - \overline{e_i^{h,l}})(e_i^{h,l} d_i^{h,l} - \overline{e_i^{h,l} d_i^{h,l}}) = 0$$

$$\Rightarrow b = -\frac{\sum_l (e_i^{h,l} - \overline{e_i^{h,l}}) \cdot (e_i^{h,l} \cdot d_i^{h,l} - \overline{e_i^{h,l} \cdot d_i^{h,l}})}{\sum_l (e_i^{h,l} - \overline{e_i^{h,l}})^2}$$

Unlike b that works on all N_i persons, our a and focal length c rely on the best-represented person instead. The main reason is that 3D key point estimator f_{kd3} is more likely to generate accurate estimation on the best-represented person as it provides more 2D visual cues. Mathematically, we have:

$$l* = \arg\max_l \quad area(l) \cdot key(l) \cdot prob(l)$$

where $area(l)$ denotes the size of the l-th person measured by the size of its tightest bounding box. $key(l)$ measures the proportion of 2D key points that have been detected in person l and $prob(l)$ is its probability of classifying as "person".

After obtaining the best represented person $l*$, we then can then have $\max(d_i^{r,l*}) = \frac{1}{a(\min(d_i^{l*})+b)}$ and $\min(d_i^{r,l*}) = \frac{1}{a(\max(d_i^{l*})+b)}$, where $\min(d_i^{r,l*})$ and $\max(d_i^{r,l*})$ denotes the the minimum and maximum absolute depth value on $l*$-th person's mask. Similarly, $\min(d_i^{l*})$ and $\max(d_i^{l*})$ denotes the the minimum and maximum relative depth value on $l*$-th person's mask instead. And Then we have:

$$\begin{aligned} \max(d_i^{r,l*}) - \min(d_i^{r,l*}) &= s_i^{d,l*} \\ &= \frac{1}{a(\min(d_i^{l*}) + b)} - \frac{1}{a(\max(d_i^{l*}) + b)} \end{aligned}$$

Then we have:

$$s_i^{d,l*} \cdot a = \frac{1}{(\min(d_i^{l*}) + b)} - \frac{1}{(\max(d_i^{l*}) + b)}$$

$$\Rightarrow a = (\frac{1}{(\min(d_i^{l*}) + b)} - \frac{1}{(\max(d_i^{l*}) + b)}) / s_i^{d,l*}$$

*Corresponding author.

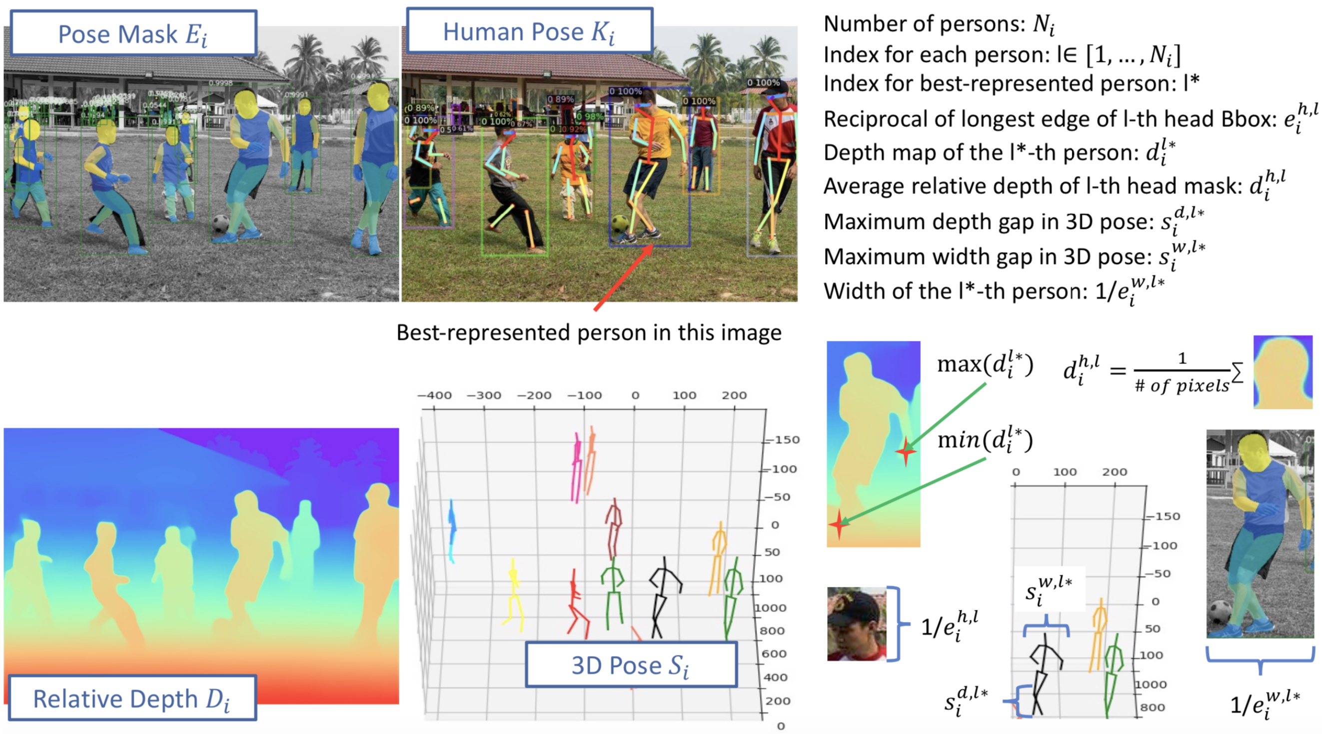

Figure 1. Definitions of used variables.

Given a and b, we can easily obtain the absolute depth by $D_i^r = \frac{1}{a(D_i - b)}$. Denoting the average absolute depth of $l*$-th person as $d_i^{r,l*}$, we have:

$$\frac{c}{d_i^{r,l*}} = \frac{1}{e_i^{w,l*} * s_i^{w,l*}}$$

which gives us an estimation of $c = d_i^{r,l*} / (s_i^{w,l*} \cdot e_i^{w,l*})$

2. Model Structure and Training Procedure

2.1. Model Structure

We provide more details for each module of our ESCNet in this section. Specifically, Tab. 1 describes detailed structure of feature extractor, e.g. head, view field and scene feature extractor, in geometry and scene parsing model f_{gp} and f_{sp}. As we described in our main paper, they all share the same ResNet50 [2] backbone. Tab. 2 and Tab. 3 shows the detailed encoder-decoder structure in f_{gp} and f_{sp} respectively. In Tab. 4, we provide details about the *binary prediction module* in f_{sp}, which includes MLP and in-out feature extractor.

2.2. Training Procedure

On GazeFollow [4], we train our ESCNet from scratch for 40 epochs, with learning rate set to 0.00025 and batch size of 92. As for VideoAttentionTarget [1] dataset, we initialize our ESCNet with the above mentioned model that is pre-trained on GazeFollow and then finetune it for 10 epochs with learning rate of 0.00025 and bath size of 92. As for out-of-frame prediction, we again initialize ESCNet with model that finetuned on VideoAttentionTarget and then only update parameters of binary prediction module. We use ADAM [3] as our optimiser and set λ to 10 according the performance on validation set.

feature extractor in f_{gp}, f_{sp}		
Layer type	Dimensions	Output (h,w,c)
ResNet-50	-	7,7,2048
Deconv2D*	1×1, stride 2	14,14,512

Table 1. Summary of the feature extractors for head, view field and scene. Layer marked with $*$ are followed by batch normalization and Relu.

3. Analysis on Our Method

3.1. 3D Visualization

We visualize our results on GazeFollow [4] in Fig. 2. Compared to figures in our main paper that mainly in 2D, we would like to highlight the 3D property of our proposed method instead in supplementary. From left to right, we visualize the original RGB with the target person and 2D

encoder-decoder in f_{gp}		
Layer type	Dimensions	Output (h,w,c)
Conv2D*	1×1, stride 1	14,14,512
Conv2D*	1×1, stride 1	14,14,256
Deconv2D*	3×3, stride 2	28,28,128
Deconv2D*	3×3, stride 4	112,112,16
Deconv2D*	4×4, stride 2	224,224,1
Deconv2D	1×1, stride 1	224,224,1

Table 2. Summary of the encoder-decoder in f_{gp}. Layer marked with $*$ are followed by batch normalization and Relu.

encoder-decoder in f_{sp}		
Layer type	Dimensions	Output (h,w,c)
Conv2D*	1×1, stride 1	14,14,1024
Conv2D*	1×1, stride 1	14,14,512
Deconv2D*	3×3, stride 2	30,30,256
Deconv2D*	3×3, stride 2	61,61,128
Deconv2D*	4×4, stride 1	64,64,1
Deconv2D	1×1, stride 1	64,64,1

Table 3. Summary of the encoder-decoder in f_{sp}. Layer marked with $*$ are followed by batch normalization and Relu.

feature extractor and MLP in binary prediction module		
Layer type	Dimensions	Output (h,w,c)
Conv2D*	1×1, stride 1	14,14,512
Conv2D*	1×1, stride 1	14,14,1
Linear	196×1	1

Table 4. Summary of binary prediction module in f_{sp}. Layer marked with $*$ are followed by batch normalization and Relu.

ground truth highlighted, generated 3D point clouds P_i, ground truth annotations in 3D, front-most points in 3D, initial heatmap in 3D and our final prediction in 3D. Note that in GazeFollow we have 10 annotations for each given person, so we visualize both the averaged position over 10 annotations and individual annotations in 3D, with red and green.

We can see that firstly, our proposed method can generate satisfactory 3D point clouds P_i with single image (the second column). Secondly, the individual ground truth annotations capture the gaze target well but the averaged position can sometimes be very noisy (the third column). The forth column, or the one reflects the front-most points, can almost always captures the occlusion relationship by excluding the occluded 3D points w.r.t. the given person. Finally, we can see that out initial heatmap and final prediction can gradually narrow down the target area and provide good estimation about the gaze fixation in last two columns.

Metric	X (pix.)						
	1	2	5	10	20	50	100
Recall	1.0	3.5	13.7	30.2	49.8	73.8	90.1
	Normalized Distance						
	.002	.005	.01	.02	.05	.1	.2
Recall	1.1	5.3	13.8	30.2	54.9	72.3	95.8

Table 5. Recall rate of front-most points on GazeFollow test set.

Metric	2D		3D	
	Dist.(pix.)	Ang.($^\circ$)	Dist.(mm)	Ang.($^\circ$)
	141.2	17.7	3026	25.4

Table 6. Evaluation of the highest probability point in A_i^* w.r.t. the averaged ground truth on GazeFollow test set.

3.2. Reliability of Our Representations

To evaluate how reliable our representations are, we design several evaluation metrics on test set of GazeFollow in below.

We firstly report the percentage of annotations that falls into the range of front-most points, which can be regarded as recall of front-most points. For instance, if 2 annotations out of 10 in one test image are within X-pixel distance of any front-most points in this image, then the recall rate is 20%. Since image size varies in GazeFollow, we also normalize them to 1 and report the recall rate of our front-most points in normalized images. The higher the recall rate is, the more reliable our front-most points are.

We report the recall rate in percentage in Tab.5. As a reference, the average normalized distance in terms of human annotations are reported to be 0.096 [4]. In comparison, we can see that our front-most points can almost always capture the ground truth, e.g. we can capture 72.3% of ground truth when the normalized distance threshold is set to 0.1. We also notice that 10% of ground truth annotations are more than 100 pixels away from any front-most points. We visualize such cases in Fig. 3. As can be seen in this figure, the ground truth annotations can sometimes be very noisy thus are far away from our generated front-most points. For instance, one cannot see the salad in the ball due to occlusions but some annotations actually fall into such area.

Our second setting focuses more on the averaged locations. Rather than measuring over 10 annotations independently, we use their averaged location as ground truth. For each test image, we first obtain the averaged location over 10 annotations in 2D and then map it to 3D. In the meantime, we get the 2D point with the highest probability in A_i^* and map it to 3D as well. Later, we report their 1) average distance in 2D image space 2) average distance in 3D space 3) angular gap in 2D image space and finally 4) angular gap in 3D space.

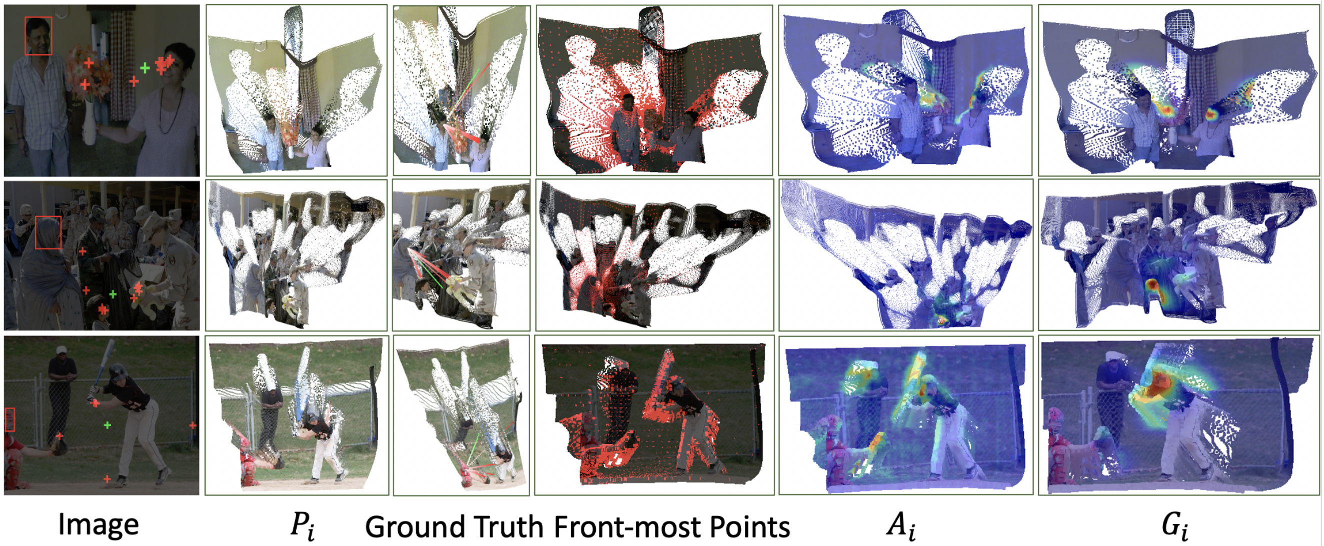

Figure 2. From left to right, we visualize the original RGB with the target person and 2D ground truth highlighted, generated point clouds P_i, ground truth annotations, front-most points and initial heatmap and our final prediction in 3D. We highlight individual annotations in red and averaged position in green.

Figure 3. We visualize the ground truth annotations and our front-most points in this figure. We highlight individual annotations in red and averaged position in green.

In Tab. 6 we demonstrate the results of our second setting. The angular error of human annotations is 11.0 [4] while we achieve 17.7 with generated A_i^* in 2D image space. We also report our numbers in 3D, which are missing in literature. Though we have about 3m distance w.r.t. average ground truth location in 3D, we can see that it is also mainly due to the noisy averaged location (see third column in Fig. 2). Again, we would like to note that the averaged location can be noisy (see Fig. 2 and Fig. 3) and our measurements on averaged location can be less meaningful.

References

[1] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5406, 2020. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

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

[4] Adrià Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? In *NIPS*, 2015. 2, 3, 4