Supplementary Material for
Coarse-to-Fine Gaze Redirection with Numerical and Pictorial Guidance

This supplementary document provides additional results supporting the claims of the main paper.

Firstly, we show the network architecture in Table. 1 and Table. 2. Secondly, the Gazemaps corresponding to numeric angles are shown in Fig. 1. Then, we show more gaze redirection results in Fig. 2, Fig. 3 and Fig. 4 to validate the effectiveness and superiority of CFGR.

1. Network Architecture

Here are some notations should be noted: h: height of input images; w: width of the input images; C: number of output channels; K: size of kernels; S: strides of kernels; IN: instance normalization; lRelu: leaky ReLu; BS: bilinear samples; f: flow field. In our experiments, $h = 64, w = 64$. We use "SAME" padding for all convolutional layers.

Tab. 1 shows the network architecture of Encoder-Decoder with NPG module. More details about NPG module can be found in the full paper. Tab. 2 shows the network architecture of Generator and Discriminator.

Module	Input Shape	Layer Information	Output Shape	Comments
encoder	$(h, w, 3)$	Reshape, Concat	$(h, w, 4)$	input image
	(1)			head pose
	$(h, w, 4)$	CONV-(C32, K7×7, S1×1),IN,lRelu	$(h, w, 32)$	
	$(h, w, 32)$	CONV-(C64, K4×4, S2×2),IN,lRelu	$(\frac{h}{2}, \frac{w}{2}, 64)$	
	$(\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(C128, K4×4, S2×2),IN,lRelu	$(\frac{h}{4}, \frac{w}{4}, 128)$	
	$(\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(C256, K4×4, S2×2),IN,lRelu	$(\frac{h}{8}, \frac{w}{8}, 256)$	feature
decoder	$(\frac{h}{8}, \frac{w}{8}, 256)$	Reshape, Concat	$(\frac{h}{8}, \frac{w}{8}, 262)$	feature of encoder
	(2)			angle difference vector
	$(h, w, 4)$			gazemaps
	$(\frac{h}{8}, \frac{w}{8}, 262)$	DECONV-(C128,K4×4,S2×2),IN,lRelu	$(\frac{h}{4}, \frac{w}{4}, 128)$	
	$(\frac{h}{4}, \frac{w}{4}, 128)$	DECONV-(C64,K4×4,S2×2),IN,lRelu	$(\frac{h}{2}, \frac{w}{2}, 64)$	
	$(\frac{h}{2}, \frac{w}{2}, 64)$	DECONV-(C32,K4×4,S2×2),IN,lRelu	$(h, w, 32)$	
	$(h, w, 32)$	CONV-(C2, k1×1, S1×1)	$(h, w, 2)$	f
	$(h, w, 2)$	BS(input, f)	$(h, w, 3)$	bilinear samples

Table 1. The architectures of Encoder and Decoder in CFGR. The encoder will concatenate the source image and head pose as input and decoder will concatenate the encoded features and NPG as input. The guidance consists of the angle differences vector, the corresponding gazemaps of input angle and target angle. This module aims to learn the flow field to warp the input for learning the spatial transformation of angle redirection.

Module	Input Shape	Layer Information	Output Shape	Comments
Generator	$(h, w, 6)$	Reshape, Concat	$(h, w, 13)$	image
	(2)			angle difference vector
	(1)			head pose
	$(h, w, 4)$			gazemaps
	$(h, w, 13)$	CONV-(C32, K7×7, S1×1),IN,lRelu	$(h, w, 32)$	
	$(h, w, 32)$	CONV-(C64, K4×4, S2×2),IN,lRelu	$(\frac{h}{2}, \frac{w}{2}, 64)$	
	$(\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(C128, K4×4, S2×2),IN,lRelu	$(\frac{h}{4}, \frac{w}{4}, 128)$	
	$(\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(C256, K4×4, S2×2),IN,lRelu	$(\frac{h}{8}, \frac{w}{8}, 256)$	
	$(\frac{h}{8}, \frac{w}{8}, 256)$	DECONV-(C128,K4×4,S2×2),IN,lRelu	$(\frac{h}{4}, \frac{w}{4}, 128)$	
	$(\frac{h}{4}, \frac{w}{4}, 256)$	DECONV-(C64,K4×4,S2×2),IN,lRelu	$(\frac{h}{2}, \frac{w}{2}, 64)$	cascaded feature maps
	$(\frac{h}{2}, \frac{w}{2}, 128)$	DECONV-(C32,K4×4,S2×2),IN,lRelu	$(h, w, 32)$	cascaded feature maps
	$(h, w, 32)$	CONV-(C3, K1×1, S1×1), Tanh	$(h, w, 3)$	results
Discriminator	$(h, w, 3)$	CONV-(C64, K4×4, S2×2),lRelu	$(\frac{h}{2}, \frac{w}{2}, 64)$	
	$(\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(C128, K4×4, S2×2),lRelu	$(\frac{h}{4}, \frac{w}{4}, 128)$	
	$(\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(C256, K4×4, S2×2),lRelu	$(\frac{h}{8}, \frac{w}{8}, 256)$	
	$(\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(C128,K4×4,S2×2),lRelu	$(\frac{h}{16}, \frac{w}{16}, 128)$	
	$(\frac{h}{16}, \frac{w}{16}, 128)$	CONV-(C64,K4×4,S2×2),lRelu	$(\frac{h}{32}, \frac{w}{32}, 64)$	
	$(\frac{h}{32}, \frac{w}{32}, 64)$	CONV-(C1, K4×4, S2×2)	$(\frac{h}{64}, \frac{w}{64}, 1)$	logits
	$(\frac{h}{32}, \frac{w}{32}, 64)$	CONV-(C2, K4×4, S2×2), GAP	$(2, 1)$	reconstruct angle

Table 2. Generator and Discriminator Architecture. Our Generator employs Unet [2] architecture. We use spectral norm [1] in every discriminator layers, which do not need extra normalized layer. More information can be found in our full paper.

2. Gazemaps Corresponding to Different Angles

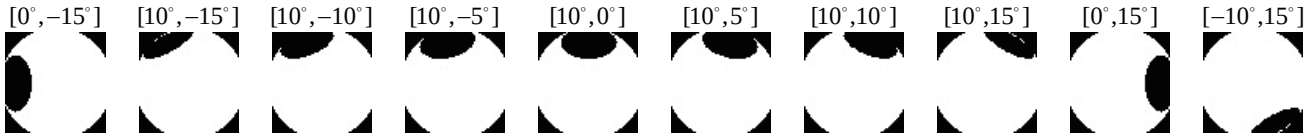

Figure 1. Gazemaps for different angles. More details can be found in Section 3.1.

3. More Gaze Redirection Results

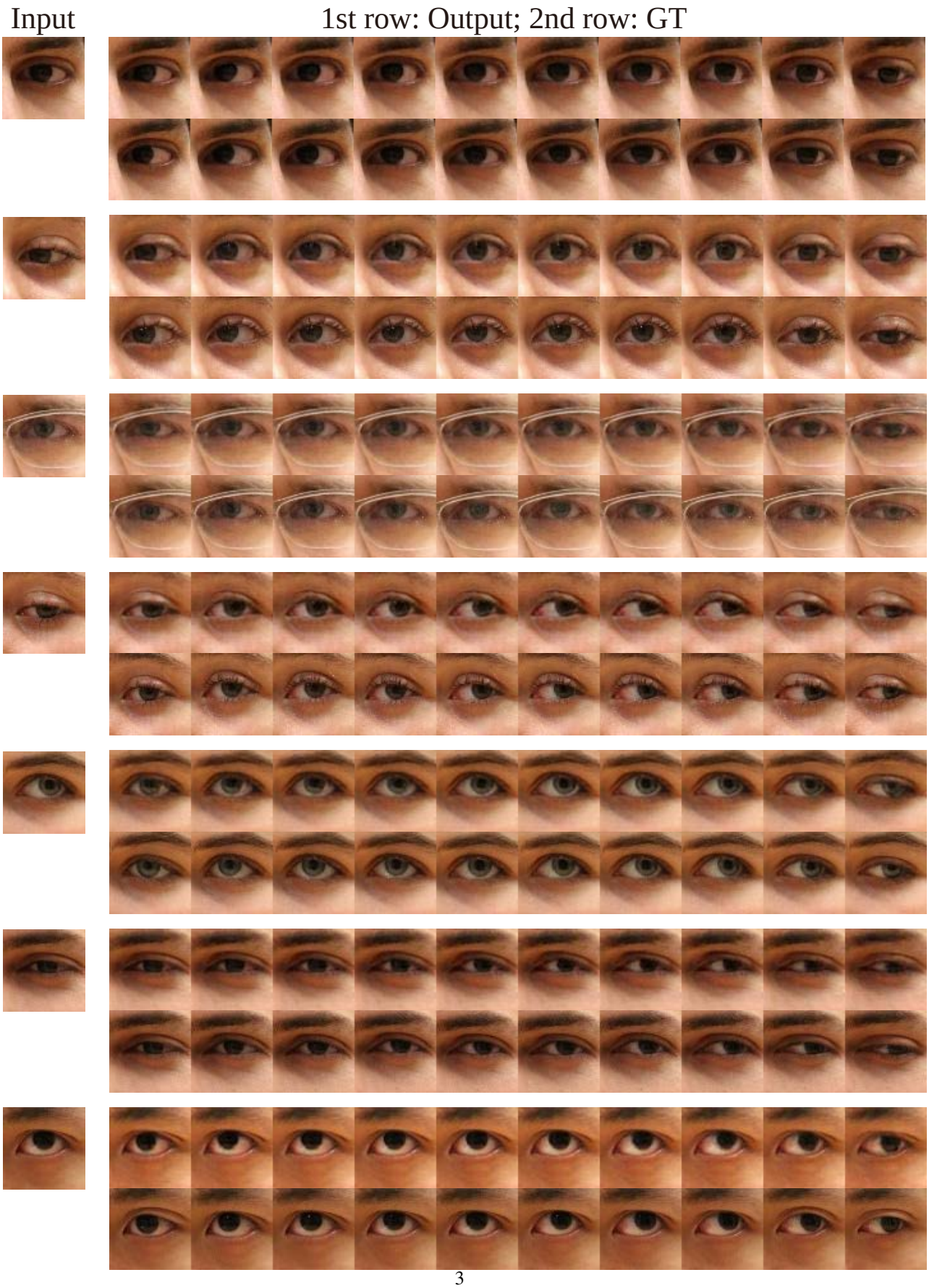

Figure 2. More high-quality gaze redirection results of CFGR.

4. More Gaze Redirection Results

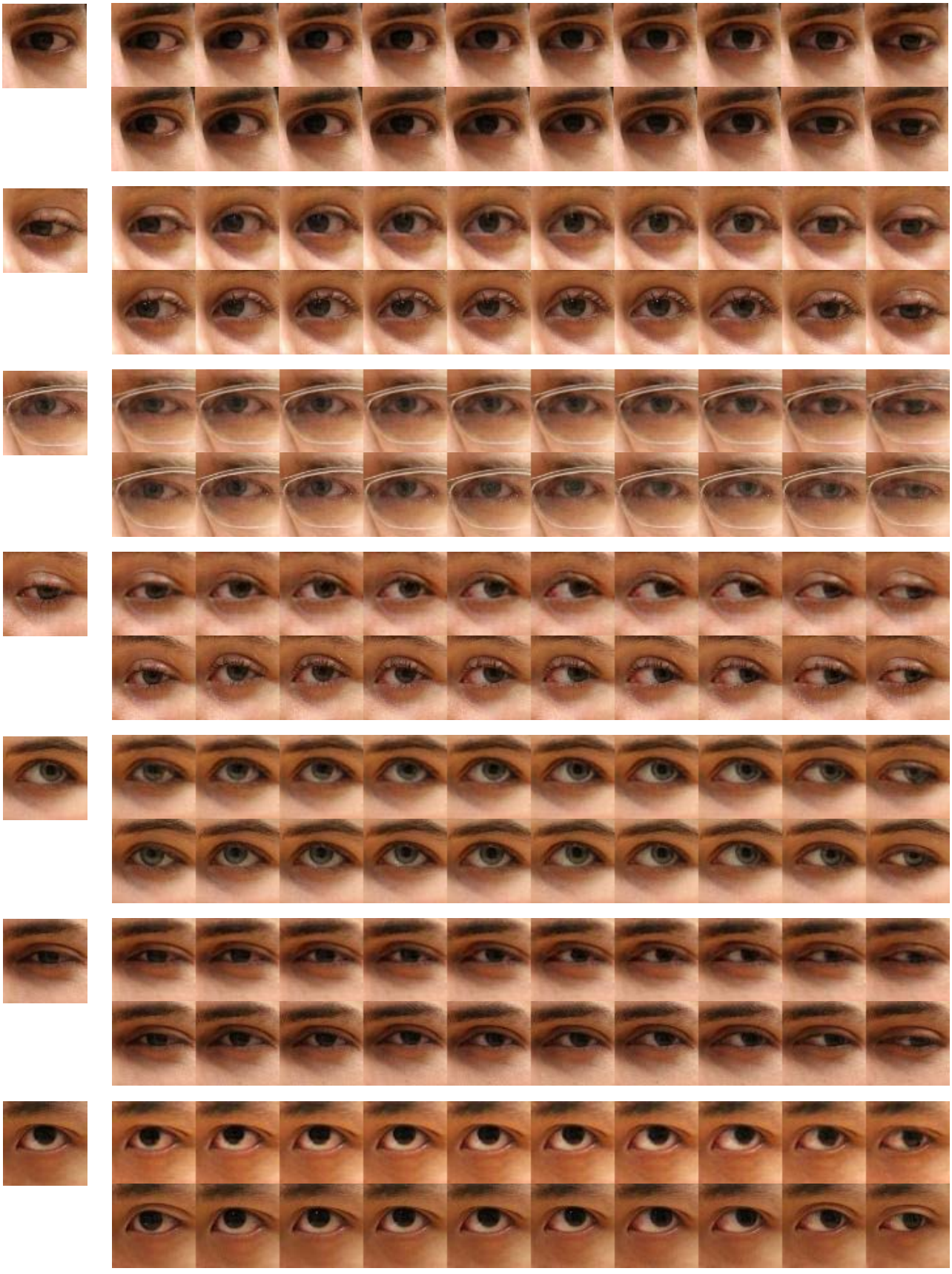

Figure 3. More high-quality gaze redirection results of CFGR.

5. More Binocular Gaze Redirection Results

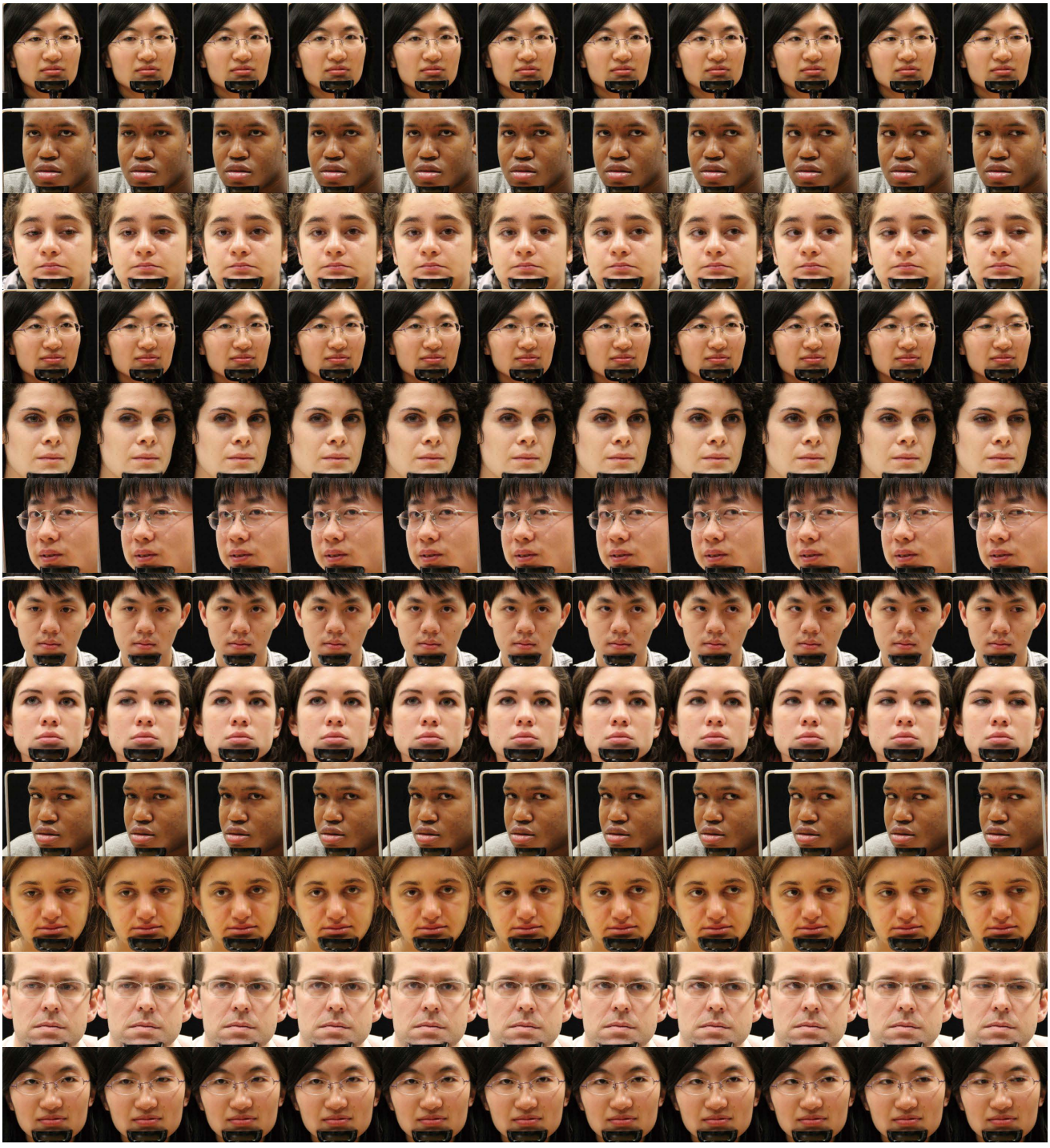

Figure 4. More high-quality binocular gaze redirection results of CFGR. For each row, the 1st column shows the input image, while the other columns show the results of the gaze redirection process with respect to different angles.

References

[1] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2