One-Click Upgrade from 2D to 3D: Sandwiched RGB-D Video Compression for Stereoscopic Teleconferencing
Supplementary Material

Yueyu Hu[1] Onur G. Guleryuz[2] Philip A. Chou Danhang Tang[2] Jonathan Taylor[2],
Rus Maxham[2] Yao Wang[1]
[1]Tandon School of Engineering, New York University [2]Google LLC
{yyhu, yaowang}@nyu.edu pachou@ieee.org
{oguleryuz, danhangtang, jontaylor, rrrus}@google.com

1. Architecture Details

In this section, we present more details on the neural network architecture used in our method. As mentioned in the main paper, our method is based on the sandwiched codec, which has a neural preprocessor and a postprocessor. Both neural processors are constructed using the U-Net architecture [1] and the multi-layer perceptron (MLP) architecture. As shown in Fig. 3 from the main paper, the preprocessor is composed of one U-Net (see architecture details in Table 1) and two MLPs (see architecture details in Table 2 with $C = 6$ for each of them). The post-processor is composed of one U-Net (see architecture details in Table 1) and one MLP (see architecture details in Table 2 with $C = 12$).

2. Visualization of Neural Codes

We visualize the learned neural code generated by the preprocessor, with a low-bit-rate model (in Fig. 1) and a high-bit-rate model (in Fig. 2), respectively. As described in the main paper, we organize the 12 latent channels into 4 groups, each group containing 3 channels, corresponding to the Y', U', and V' channels of the YUV color space. As shown, at higher bit-rates, the preprocessor learns to maintain more detailed information by fully utilizing the 12 neural code channels. In contrast, at lower bit-rates, the preprocessor learns to compress the information by compressing the input stereo RGB-D information into fewer channels and produce high frequency modulation in the compact latent channels. The resulting neural code channels are thus more sparse and different from the human perceptable RGB-D information.

3. Extension: Relightability

Since decoder side relightability is sometimes desired in a scene transmission pipeline. In this section, we demonstrate a simple extension of our method to support decoder-side relighting by transmitting normal maps together with the RGB-D signal in our sandwiched codec. We qualitatively evaluate the extension setting by relighting the transmitted 3D representation. Results shown in Fig. 3 demonstrate that our method can be easily extended to a relighting rendering pipeline and maintains better performance.

References

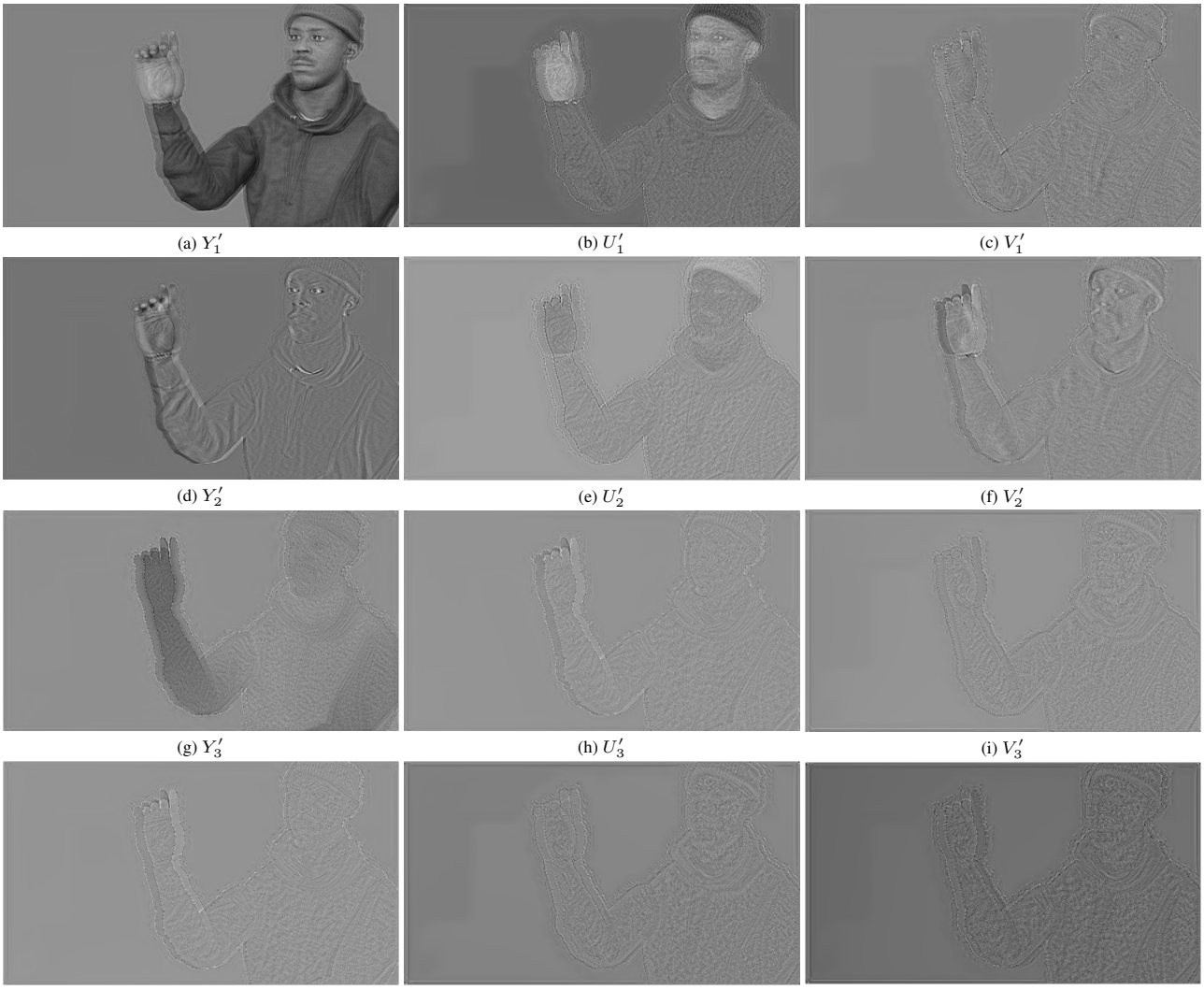
[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

Table 1. Architecture details for the U-Net used in our method.

Layer	Type	Input Source	Input Shape	Output Shape	Kernel	Stride
1	Convolution	-	(12, H, W)	(64, H, W)	3x3	1
2	Convolution	1	(64, H, W)	(64, H, W)	3x3	1
3	Convolution	2	(64, H, W)	(128, H/2, W/2)	5x5	2
4	Convolution	3	(128, H/2, W/2)	(128, H/2, W/2)	3x3	1
5	Convolution	4	(128, H/2, W/2)	(256, H/4, W/4)	5x5	2
6	Convolution	5	(256, H/4, W/4)	(256, H/4, W/4)	3x3	1
7	Convolution	6	(256, H/4, W/4)	(512, H/8, W/8)	5x5	2
8	Convolution	7	(512, H/8, W/8)	(512, H/8, W/8)	3x3	1
9	Convolution	8	(512, H/8, W/8)	(512, H/16, W/16)	5x5	2
10	Convolution	9	(512, H/16, W/16)	(512, H/16, W/16)	3x3	1
11	Transposed Conv.	10	(512, H/16, W/16)	(512, H/16, W/16)	5x5	2
12	Concatenation	11, 8	2 x (512, H/8, W/8)	(1024, H/8, W/8)	-	-
13	Convolution	12	(1024, H/8, W/8)	(512, H/8, W/8)	3x3	1
14	Transposed Conv.	13	(512, H/8, W/8)	(256, H/4, W/4)	5x5	2
15	Concatenation	14, 6	2 x (256, H/4, W/4)	(512, H/4, W/4)	-	-
16	Convolution	15	(512, H/4, W/4)	(256, H/4, W/4)	3x3	1
17	Transposed Conv.	16	(256, H/4, W/4)	(128, H/2, W/2)	5x5	2
18	Concatenation	17, 4	2 x (128, H/2, W/2)	(256, H/2, W/2)	-	-
19	Convolution	18	(256, H/2, W/2)	(128, H/2, W/2)	3x3	1
20	Transposed Conv.	19	(128, H/2, W/2)	(64, H, W)	5x5	2
21	Concatenation	20, 2	2 x (64, H, W)	(128, H, W)	-	-
22	Convolution	21	(128, H, W)	(64, H, W)	3x3	1
23	Convolution	22	(64, H, W)	(12, H, W)	3x3	1

Table 2. Architecture details for the multi-layer perceptron (MLP) used in our method.

Layer	Type	Input Source	Input Shape	Output Shape	Kernel	Stride
1	Convolution	-	(C, H, W)	(512, H, W)	1x1	1
2	Convolution	1	(512, H, W)	(C, H, W)	1x1	1



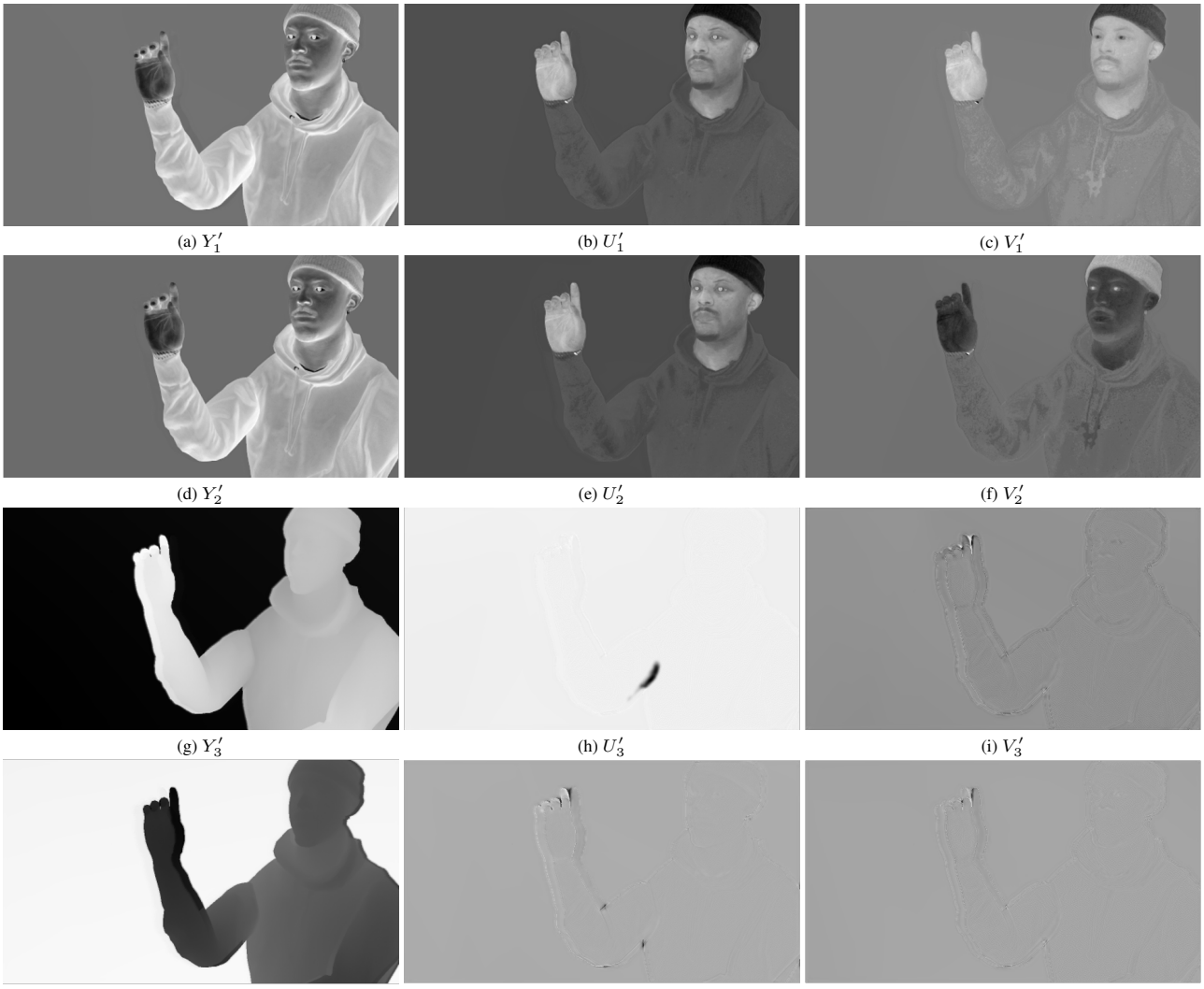
(a) Y_1' (b) U_1' (c) V_1'

(d) Y_2' (e) U_2' (f) V_2'

(g) Y_3' (h) U_3' (i) V_3'

(j) Y_4' (k) U_4' (l) V_4'

Figure 1. Visualization of the latent channels generated by the preprocessor with a low bit-rate setting.



(a) Y_1' (b) U_1' (c) V_1'

(d) Y_2' (e) U_2' (f) V_2'

(g) Y_3' (h) U_3' (i) V_3'

(j) Y_4' (k) U_4' (l) V_4'

Figure 2. Visualization of the latent channels generated by the preprocessor with a high bit-rate setting.

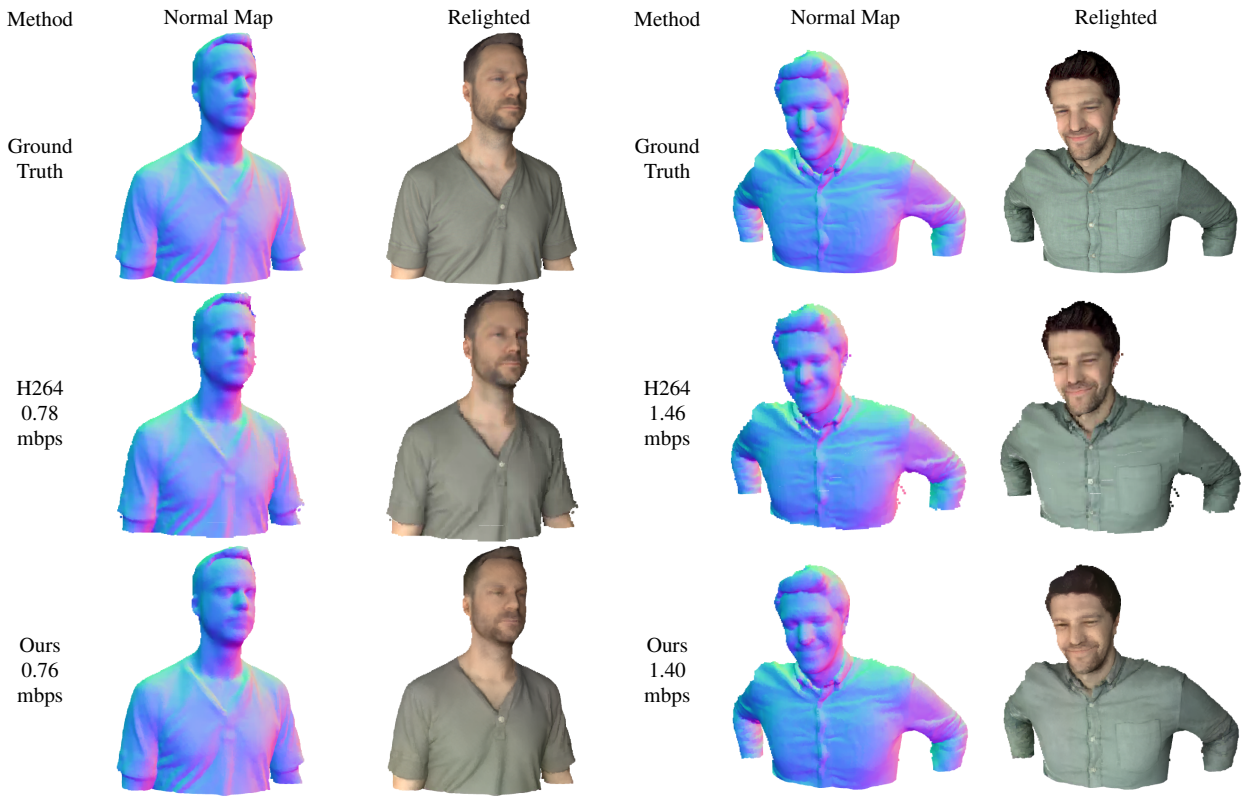


Figure 3. Qualitative results with *The Relightable* dataset showing the relightability of the decoded signal in the extension setting.