WAVELET SYNTHESIS NET FOR DISPARITY ESTIMATION TO SYNTHESIZE DSLR CALIBRE BOKEH EFFECT ON SMARTPHONES – SUPPLEMENTARY MATERIALS

Chenchi Luo*, Yingmao Li*, Kaimo Lin*, George Chen*,
Seok-Jun Lee*, Jihwan Choi+, Youngjun Francis Yoo*, Michael O. Polley*
*Samsung Research America
+Samsung Electronics
{eric.luo, timothy.li, kaimo.lin, george.chen}@samsung.com

1 WSN Detailed Topology

Table. 1 shows the detailed topology of the WSN. There are four columns in the table. The first column shows the module name. Each module contains one or more layers. The second column shows the input feature maps for each layer inside the module. The format is the feature maps name followed by dimensions in (H,W,C) order. The third column shows the output feature maps for each layer. The fourth column shows the layer type for each individual layer. In this paper, the depthwise convolution layer is defined by six sequential layers: 3×3 separable convolution, batch normalization, ReLU, 1×1 pointwise convolution, batch normalization, ReLU. For the normalized correlation layer, our baseline direction search window is $[-20, -18, -16, -14, -12, -10, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 4]$ and our orthogonal baseline direction search window is $[-4, -2, -1, 0, 1, 2, 4]$. Therefore, our search window size is 18×7 or 126 elements. As a result, the feature maps output channel is also 126 for the normalized correlation layer.

The input and output spatial resolutions are (1536,2048). The number of parameters is only 677 thousand. The corresponding total number of multiply-and-accumulate (MAC) is 27 billion.

Table 1: WSN network topology.

Module	Inputs	Outputs	Layer(s)
Wavelet0	L (1536,2048,3)	wavelet0[L] (768,1024,3) wavelet0[H] (768,1024,9)	wavelet
Wavelet0_1	R (1536,2048,3)	wavelet0_1[L] (768,1024,3) wavelet0_1[H] (768,1024,9)	wavelet
Conv1	wavelet0[L] (768,1024,3)	conv1[0] (768,1024,32)	depthwise conv
Conv1_1	wavelet0_1[L] (768,1024,3)	conv1_1[0] (768,1024,32)	depthwise conv
Conv1_2	wavelet0[H] (768,1024,9)	conv1_2[0] (768,1024,96)	depthwise conv
Wavelet1	conv1[0] (768,1024,32)	wavelet1[L] (384,512,32) wavelet1[H] (384,512,96)	wavelet
Wavelet1_1	conv1_1[0] (768,1024,32)	wavelet1_1[L] (384,512,32) wavelet1_1[H] (384,512,96)	wavelet
Conv2	wavelet1[L] (384,512,32)	conv2[0] (384,512,64)	depthwise conv
Conv2_1	wavelet1_1[L] (384,512,32)	conv2_1[0] (384,512,64)	depthwise conv
Conv2_2	wavelet1[H] (384,512,96)	conv2_2[0] (384,512,192)	depthwise conv
Wavelet2	conv1[0] (384,512,64)	wavelet2[L] (192,256,64) wavelet2[H] (192,256,192)	wavelet
Wavelet2_1	conv1_1[0] (384,512,64)	wavelet2_1[L] (192,256,64) wavelet2_1[H] (192,256,192)	wavelet
Conv3	wavelet2[L] (192,256,64)	conv3[0] (192,256,128)	depthwise conv

	conv3[0] (192,256,128) conv3[1] (192,256,128)	conv3[1] (192,256,128) conv3[2] (192,256,128)	depthwise conv depthwise conv
Conv3_1	wavelet2_1[L] (192,256,64) conv3_1[0] (192,256,128) conv3_1[1] (192,256,128)	conv3_1[0] (192,256,128) conv3_1[1] (192,256,128) conv3_1[2] (192,256,128)	depthwise conv depthwise conv depthwise conv
Conv3_2	wavelet2[H] (192,256,192)	conv3_2[0] (192,256,384)	depthwise conv
Corr	conv3[2] (192,256,192) conv3_1[2] (192,256,192)	corr[0] (192,256,126)	normalized corr
Conv4	conv3[2] (192,256,192)	conv4[0] (192,256,128)	depthwise conv
Concat	conv4[0] (192,256,128) corr[0] (192,256,126)	concat[0] (192,256,254)	concat
Conv5	concat[0] (192,256,254)	conv5[0] (192,256,256)	depthwise conv
	conv5[0] (192,256,256)	conv5[0L] (96,128,256) conv5[0H] (96,128,768)	wavelet
	conv5[0L] (96,128,256)	conv5[1] (96,128,256)	depthwise conv
	conv5[1] (96,128,256)	conv5[1L] (48,64,256) conv5[1H] (48,64,768)	wavelet
	conv5[1L] (48,64,256)	conv5[2] (48,64,256)	depthwise conv
	conv5[2] (48,64,256)	conv5[2L] (24,32,256) conv5[2H] (24,32,768)	wavelet
	conv5[2L] (24,32,256)	conv5[3] (24,32,256)	depthwise conv
	conv5[3] (24,32,256) conv5[2H] (24,32,768)	conv5[4] (48,64,256)	inverse wavelet
	conv5[4] (48,64,256)	conv5[5] (48,64,256)	depthwise conv
	conv5[5] (48,64,256) conv5[1H] (48,64,768)	conv5[6] (96,128,256)	inverse wavelet
	conv5[6] (96,128,256)	conv5[7] (96,128,256)	depthwise conv
	conv5[7] (96,128,256) conv5[0H] (96,128,768)	conv5[8] (192,256,256)	inverse wavelet
	conv5[8] (192,256,256)	conv5[9] (192,256,128)	depthwise conv
iWavelet2	conv5[9] (192,256,128) conv3_2[0] (192,256,384)	iwavelet2[0] (384,512,128)	inverse wavelet
Conv6	iwavelet2[0] (384,512,128)	conv6[0] (384,512,96)	depthwise conv
iWavelet1	conv6[0] (384,512,96) conv2_2[0] (384,512,192)	iwavelet1[0] (768,1024,96)	inverse wavelet
Conv7	iwavelet1[0] (768,1024,96)	conv7[0] (768,1024,32)	depthwise conv
iWavelet0	conv7[0] (768,1024,32) conv1_2[0] (768,1024,96)	iwavelet0[0] (1536,2048,32)	inverse wavelet
Conv8	iwavelet0[0] (1536,2048,32)	flow (1536,2048,2)	conv (3×3)

2 Bokeh Comparsion with Flagship Smartphones

In this section, we show the disparity map produced by our WSN and the final rendered bokeh image with our smartphone based bokeh pipeline. We compare the result with the other four top ranked flagship smartphones on the DXOMARK mobiles bokeh leader board. We adjust the images so that they approximately have the same field of view. The images are arranged in the following order. (a) shows the disparity map produced by WSN. (b) shows our rendered bokeh image based on (a). (c) shows the bokeh image produced by the Apple iPhone 11 Pro Max. (d) shows the bokeh image produced by the Google Pixel 4 (e) shows the the bokeh image produced by the Huawei Mate30 Pro. (f) shows the the bokeh image produced by the Samsung Note10+.

Although we do not have access to the disparity map from the other smartphones, we still can infer the quality of their disparity map from the rendered bokeh image. We have labeled the depth artifacts in the bokeh image using red bounding boxes. Generally speaking, the disparity estimation of the top ranked flagship smartphones struggle at see-through holes, thin structures, foreground object boundaries, areas where the foreground object color is similar to the background color, homogeneous regions with little texture. As a result, the bokeh image produced by these smartphones cannot match the DSLR quality. In contrast, WSN demonstrates significant advantages in the scenarios where the other smartphones struggle and is able to approximate truely DSLR quality bokeh.

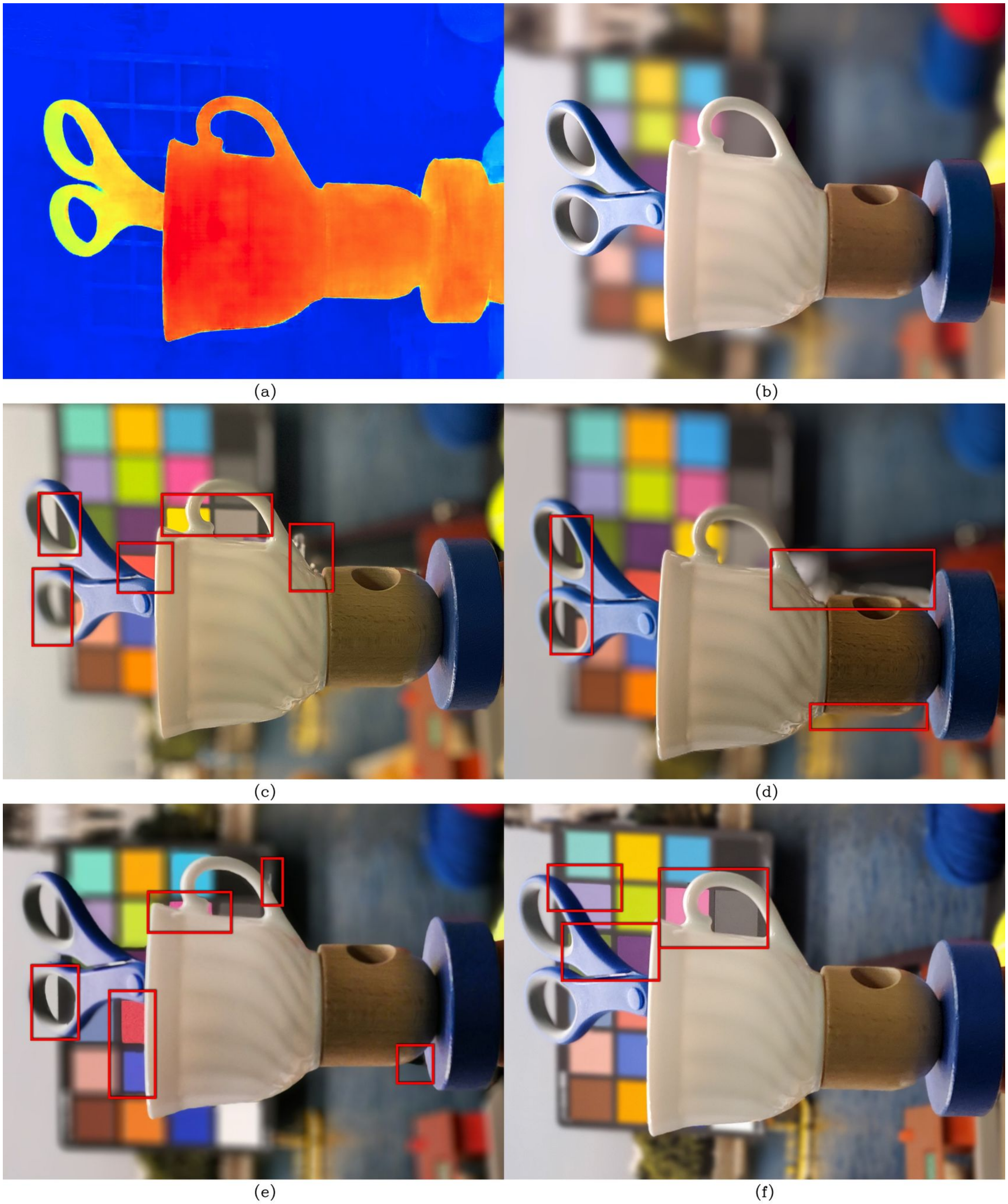


Figure 1: Bokeh comparison example 1. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)

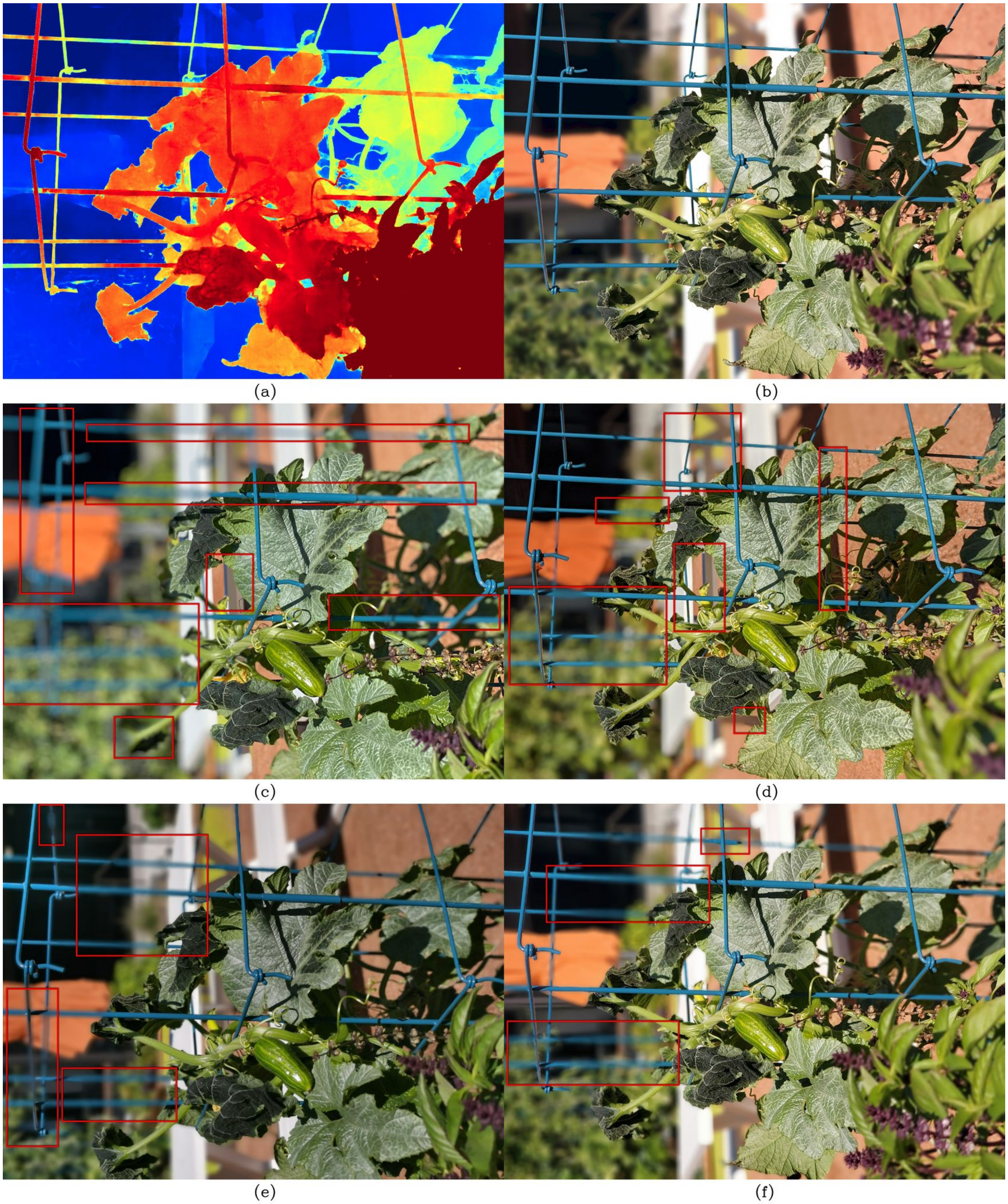


Figure 2: Bokeh comparison example 2. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)

Figure 3: Bokeh comparison example 3. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)

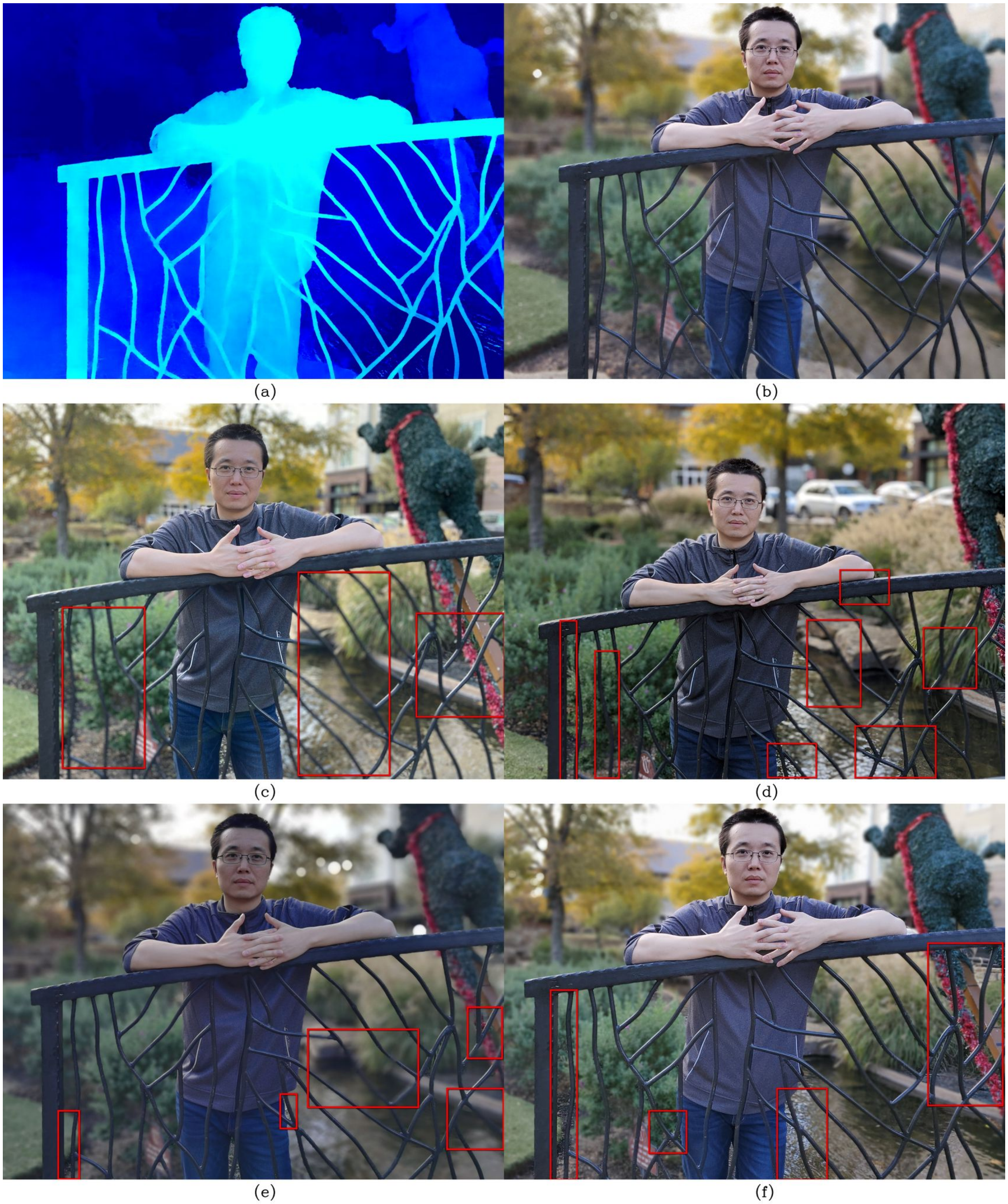


Figure 4: Bokeh comparison example 4. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)

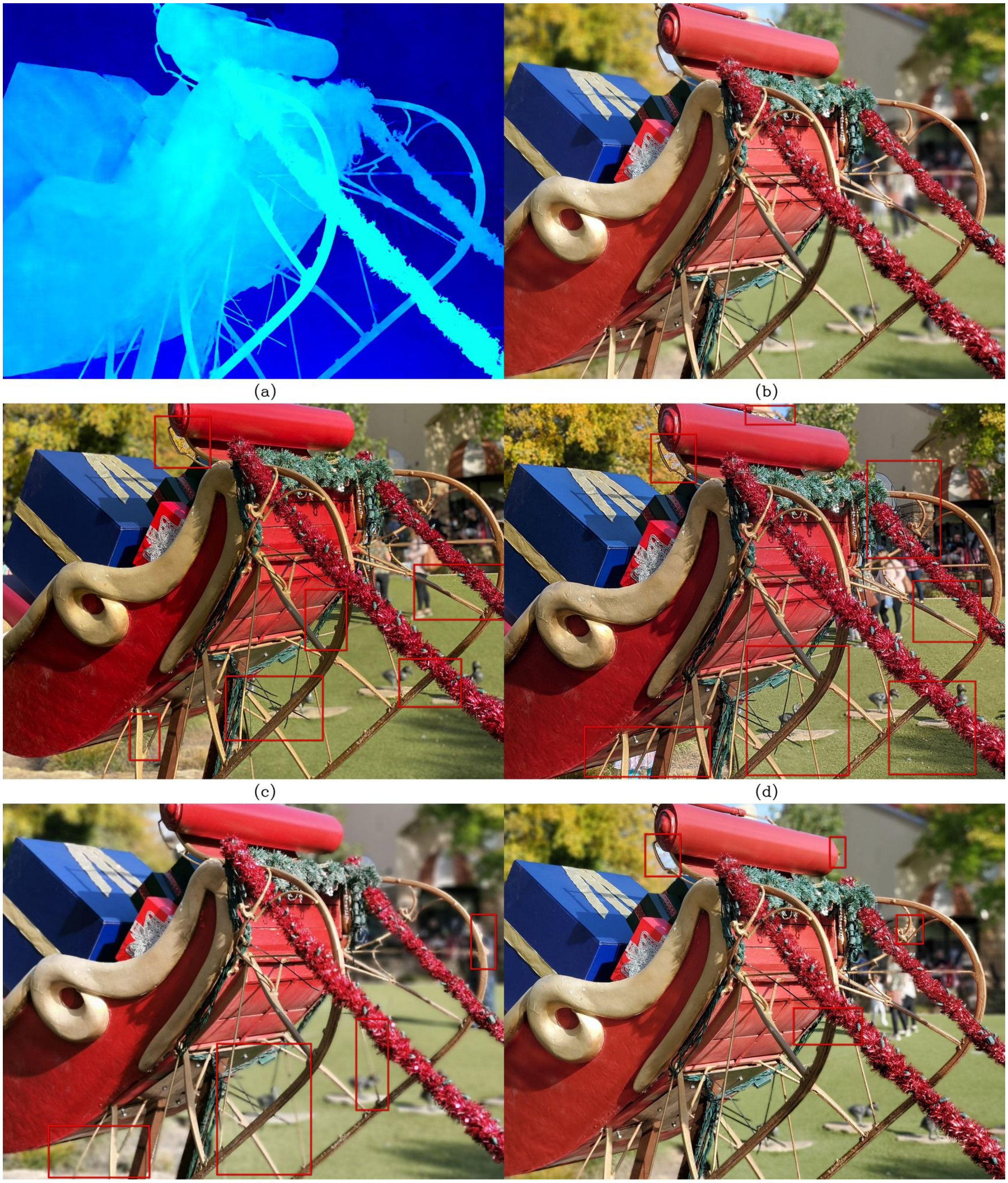
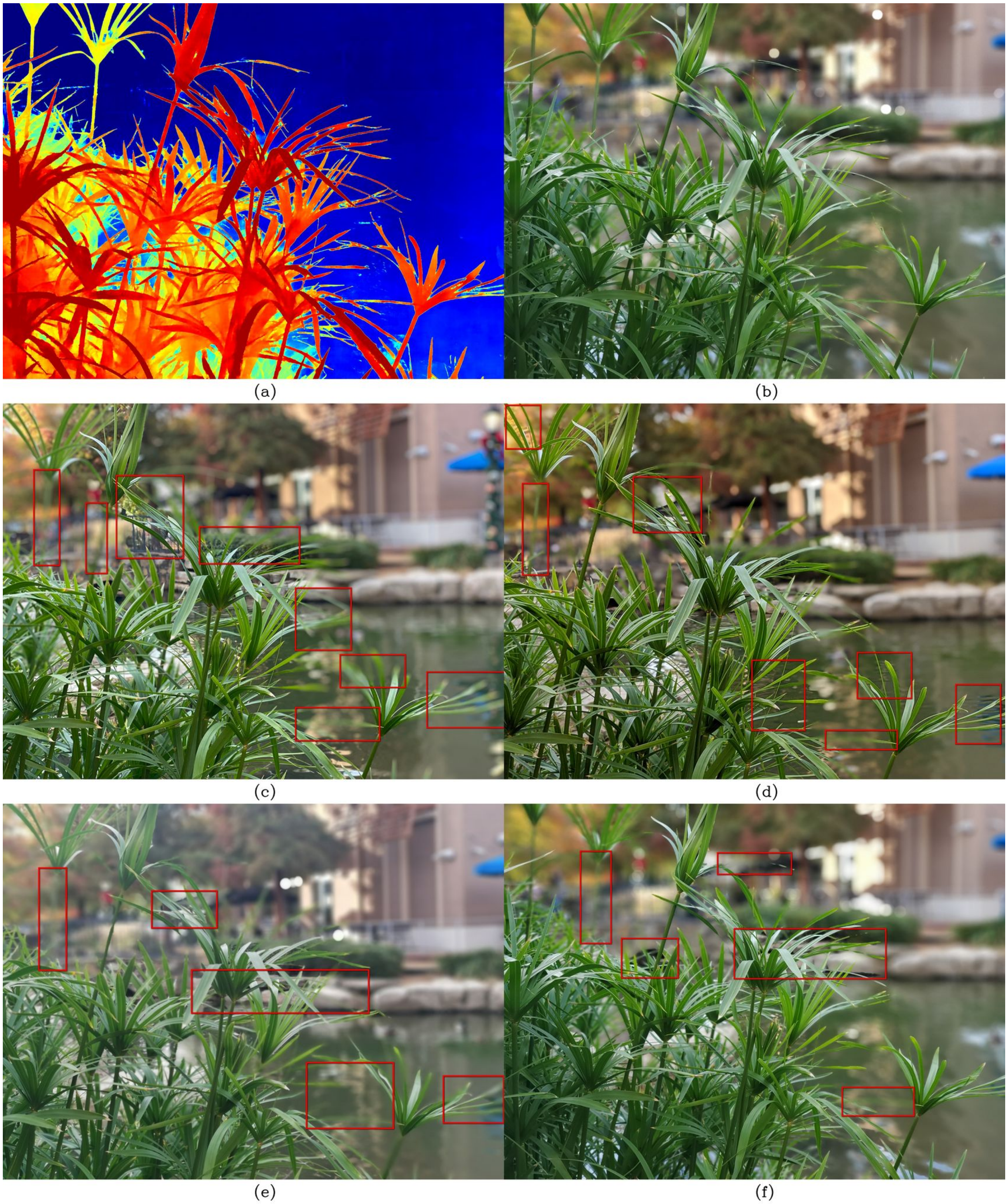


Figure 5: Bokeh comparison example 5. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)



(a) (b) (c) (d) (e) (f)

Figure 6: Bokeh comparison example 6. (a) Our disparity map. (b) Our bokeh. (c) Apple iPhone 11 Pro Max bokeh. (d) Google Pixel 4 bokeh. (e) Huawei Mate30 Pro bokeh (f) Samsung Note10+ bokeh. (Watch on computer screen recommended)