Lightweight and Fast Real-time Image Enhancement via Decomposition of the Spatial-aware Lookup Tables (APPENDIX)

Wontae Kim[1, 3], Keuntek Lee[2], and Nam Ik Cho[1, 2]

[1]IPAI, Seoul National University, Seoul, Korea
[2]Department of ECE, INMC, Seoul National University, Seoul, Korea
[3]LG Electronics, Seoul, Korea

{munte2,leekt000,nicho}@snu.ac.kr

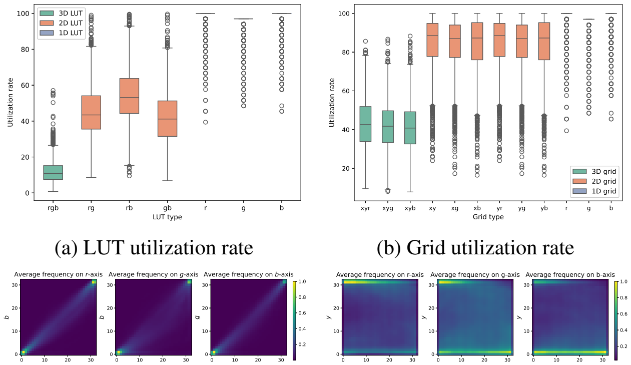
In this supplementary material, we provide the following.

- Sec. 1 Details of network structure.
- Sec. 2 Analysis for PPR10K Dataset.
- Sec. 3 Details of 2D slicing and LUT transform.
- Sec. 4 Additional quantitative and qualitative comparisons.

1. Details of Network Structure

In this section, we provide the details of our network structure as described in Tabs. 1 to 5. The backbone in Tab. 1 consists of five layers of a convolutional neural network (CNN), with m set to 8, following previous work [6]. Each generator in Tabs. 2 to 5 is composed of two fully connected (FC) layers with the insight of rank factorization, which can reduce the parameters and training difficulty [6, 10, 11]. The bilateral grid generator $H_s(\cdot)$ in Tab. 2 generates 2D bilateral grids to replace 3D bilateral grids. We decide the $D_s = 17$, $K = 6$ for fair comparison on spatial feature fusion with SABLUT [6]. The weights and biases for 2D bilateral grids are generated by the bilateral grid weight generator $H_{sw}(\cdot)$ in Tab. 3. We experimentally select the number of hidden layers for the bilateral grid weight generator as $M_{sw} = 8$. Since three 2D bilateral grids replace a 3D bilateral grid, the number of weights N_{sw} and biases N_{sb} for each channel are 3 and 1, respectively.

The LUT generator $H_t(\cdot)$ in Tab. 4 generates the singular value decomposition (SVD) components of 2D LUTs with eight singular values $N_s = 8$ as described in the main paper. Most of previous 3D LUT methods [6, 7, 9–12] have $D_t = 17$ or $D_t = 33$ vertices. When we set D_t to 17, our model does not achieve the desired performance, with a PSNR of 25.54 dB on the photo retouch task on FiveK [1]. Meanwhile, our method achieves a competitive performance of 25.76 dB with $D_t = 33$ in the same task. We decide to set D_t as 33, based on the above experiment. The 2D LUTs can be easily reconstructed by matrix multiplication, as discussed in the main paper. The LUT weight generator $H_{tw}(\cdot)$ in Tab. 5 is similar to the bilateral grid weight generator. The parameters of the LUT weight generator, $M_{tw} = 8$, $N_{tw} = 3$, and $N_{tb} = 1$, are decided for the same reason as the bilateral grid weight generator.



(a) LUT utilization rate

(b) Grid utilization rate

(c) LUT occurrence statistics

(d) Grid occurrence statistics

Figure 1. (a) and (b) are box plots of utilization rates on PPR10K dataset [7] under different dimensions of LUT and bilateral grid. Each color represents the dimension of LUT and bilateral grid. (c) and (d) are LUT visualizations of average occurrence statistics on each axis for PPR10K. The cells closer to yellow indicate more frequently accessed vertices and the cells closer to blue indicate less frequently accessed vertices.

Table 1. Details of the backbone network $B(\cdot)$ where $Conv3$ is 3×3 convolution block (stride=2, padding=1), LR is LeakyReLU and IN is instance normalization.

Layer	Output Shape
Bilinear Downsample	$3 \times 256 \times 256$
$Conv3 + LR + IN$	$m \times 128 \times 128$
$Conv3 + LR + IN$	$2m \times 64 \times 64$
$Conv3 + LR + IN$	$4m \times 32 \times 32$
$Conv3 + LR + IN$	$8m \times 16 \times 16$
$Conv3 + LR$	$8m \times 8 \times 8$
Droupout (0.5)	$8m \times 8 \times 8$
Average Pooling	$8m \times 2 \times 2$
Reshape	$32m$

Table 2. Details of the bilateral grid generator $H_s(\cdot)$, where FC is a fully connected layer. M_s, K, and D_s are the number of hidden layers, bilateral grids, and bilateral grid vertices, respectively.

Layer	Output Shape
FC	M_s
FC	$K \cdot 3 \cdot D_s^2$
Reshape	$K \times 3 \times D_s \times D_s$

Table 4. Details of the LUT generator $H_t(\cdot)$. M_t, D_t and N_S are the number of hidden layers, LUT vertices, and singular values, respectively.

Layer	Output Shape
FC	M_t
FC	$3 \cdot 3 \cdot (D_t \cdot N_S + N_S + D_t \cdot N_S)$
Reshape	$U : 3 \times 3 \times (D_t \times N_S)$ $S : 3 \times 3 \times N_S$ $V : 3 \times 3 \times (D_t \times N_S)$

Table 3. Details of the bilateral grid weight generator $H_{sw}(\cdot)$. M_{sw} is the number of hidden layers. N_{sw} and N_{sb} are the number of bilateral grid weights and biases for each channel, respectively.

Layer	Output Shape
FC	M_{sw}
FC	$K \cdot 3 \cdot (N_{sw} + N_{sb})$
Reshape	$K \times 3 \times (N_{sw} + N_{sb})$

Table 5. Details of the LUT weight generator $H_{tw}(\cdot)$. M_{tw} are the number of hidden layers. N_{tw} and N_{tb} are the number of LUT weights and biases for each channel, respectively.

Layer	Output Shape
FC	M_{tw}
FC	$3 \cdot 3 \cdot (N_{tw} + N_{tb})$
Reshape	$3 \times 3 \times (N_{tw} + N_{tb})$

2. Analysis for PPR10K Datasets

We also conduct an analysis for PPR10K on utilization rates and occurrence statistics. The utilization rate indicates how many vertices of a 3D LUT are referenced compared to generated vertices for each image, like $\frac{\#referenced\ vertices}{\#generated\ vertices} \times 100$. The occurrence statistics are estimated by counting the number of accesses for each vertex. The overall tendency is similar to the results of the analysis for FiveK in the main paper. Fig. 1a presents LUT utilization rate of PPR10K. The rate of 3D LUT is very low, and 1D LUT is saturated. As can be seen in Fig. 1c, the occurrence statistics are concentrated on the diagonal. Notably, the vertices near (1,1) and (32,32) are accessed more frequently than other diagonal vertices, compared to the FiveK dataset. The bilateral grid has a broader distribution than LUT, but it also has a similar tendency. The 3D bilateral grid is redundant, and the 1D bilateral grid is saturated, similar to the FiveK dataset, as shown in Fig. 1b. The vertices access is concentrated on as visualized in the first row and the thirty-second row in Fig. 1d.

3. Details of 2D Slicing and LUT Transform

In this section, we provide a detailed description of the 2D LUT Transform and 2D Slicing.

3.1. 2D LUT Transform

Fig. 2a illustrates the detailed operation of 2D LUT transform $Transform^c_{2D}(X_{(c,x,y)}, T^{2D})$ based on bilinear interpolation, where $T^{2D} \in \{t^c_{rg}, t^c_{rb}, t^c_{gb} | c \in \{r, g, b\}\}$ represents 2D LUTs and $X_{(c,x,y)}$ is the pixel value on (c, x, y). The 2D LUT transform comprises the following three steps.

First of all, query points for 2D LUTs are found based on input color values, which can be described as

$$\begin{aligned} P_{rg} &= (p_r, p_g) = (X_{(r,x,y)}, X_{(g,x,y)}), \\ P_{rb} &= (p_r, p_b) = (X_{(r,x,y)}, X_{(b,x,y)}), \\ P_{gb} &= (p_g, p_b) = (X_{(g,x,y)}, X_{(b,x,y)}), \end{aligned} \tag{1}$$

where P_{rg}, P_{rb}, and P_{gb} are query points for t^c_{rg}, t^c_{rb}, and t^c_{gb}, respectively.

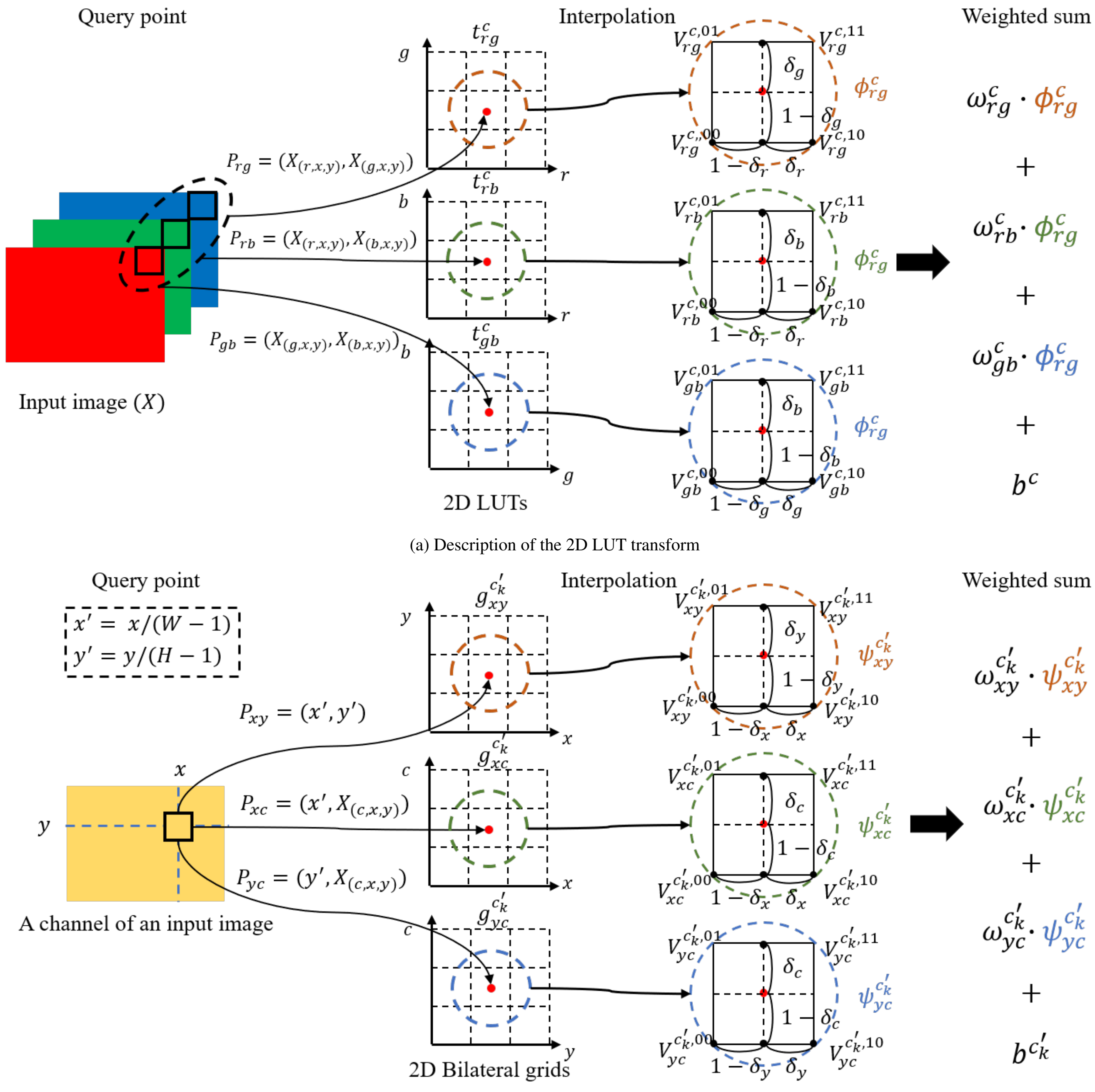
Second, the bilinear interpolation is carried out to calculate retrieved values $\phi^c_{\alpha\beta}$ from $t^c_{\alpha\beta}$ with $\alpha\beta \in \{rg, rb, gb\}$. For the interpolation operation, we find the left and right vertices of each query point on their axis. The left and right vertices can be denoted as

$$\begin{aligned} p^l_\alpha &= \lfloor p_\alpha \cdot (D_t - 1) \rfloor / (D_t - 1), \\ p^r_\alpha &= p^l_\alpha + 1/(D_t - 1), \\ p^l_\beta &= \lfloor p_\beta \cdot (D_t - 1) \rfloor / (D_t - 1), \\ p^r_\beta &= p^l_\beta + 1/(D_t - 1), \end{aligned} \tag{2}$$

where $\lfloor \cdot \rfloor$ is floor operator. The four adjacent points on a 2D LUT can be found as

$$\begin{aligned} V^{c,00}_{\alpha\beta} &= t^c_{\alpha\beta}(p^l_\alpha, p^l_\beta), \\ V^{c,10}_{\alpha\beta} &= t^c_{\alpha\beta}(p^r_\alpha, p^l_\beta), \\ V^{c,01}_{\alpha\beta} &= t^c_{\alpha\beta}(p^l_\alpha, p^r_\beta), \\ V^{c,11}_{\alpha\beta} &= t^c_{\alpha\beta}(p^r_\alpha, p^r_\beta). \end{aligned} \tag{3}$$

The retrieved values can be calculated by bilinear interpo-


Query point
Interpolation
Weighted sum
t^c_{rg}
t^c_{rb}
t^c_{gb}
$P_{rg} = (X_{(r,x,y)}, X_{(g,x,y)})$
$P_{rb} = (X_{(r,x,y)}, X_{(b,x,y)})$
$P_{gb} = (X_{(g,x,y)}, X_{(b,x,y)})$
Input image (X)
2D LUTs
$V^{c,01}_{rg}$ $V^{c,11}_{rg}$ $V^{c,00}_{rg}$ $V^{c,10}_{rg}$
δ_g $1-\delta_g$ $1-\delta_r$ δ_r
ϕ^c_{rg}
$\omega^c_{rg} \cdot \phi^c_{rg}$
+
$\omega^c_{rb} \cdot \phi^c_{rg}$
+
$\omega^c_{gb} \cdot \phi^c_{rg}$
+
b^c
$x' = x/(W-1)$
$y' = y/(H-1)$
$P_{xy} = (x', y')$
$P_{xc} = (x', X_{(c,x,y)})$
$P_{yc} = (y', X_{(c,x,y)})$
A channel of an input image
$g^{c'_k}_{xy}$
$g^{c'_k}_{xc}$
$g^{c'_k}_{yc}$
2D Bilateral grids
$\psi^{c'_k}_{xy}$ $\psi^{c'_k}_{xc}$ $\psi^{c'_k}_{yc}$
$\omega^{c'_k}_{xy} \cdot \psi^{c'_k}_{xy}$
+
$\omega^{c'_k}_{xc} \cdot \psi^{c'_k}_{xc}$
+
$\omega^{c'_k}_{yc} \cdot \psi^{c'_k}_{yc}$
+
$b^{c'_k}$

(a) Description of the 2D LUT transform

(b) Description of the 2D slicing

Figure 2. Detailed description of 2D LUT transform (a) and 2D Slicing (b) based on bilinear interpolation.

lation, which can be formulated as

$$
\begin{aligned}
\phi^c_{\alpha\beta,(x,y)} = {} & (1-\delta_\alpha)\cdot(1-\delta_\beta)\cdot V^{c,00}_{\alpha\beta} \\
& + \delta_\alpha \cdot (1-\delta_\beta)\cdot V^{c,10}_{\alpha\beta} \\
& + (1-\delta_\alpha)\cdot\delta_\beta\cdot V^{c,01}_{\alpha\beta} \\
& + \delta_\alpha\cdot\delta_\beta\cdot V^{c,11}_{\alpha\beta},
\end{aligned}
\tag{4}
$$

where $\delta_\alpha = (p_\alpha - p^l_\alpha)/(p^r_\alpha - p^l_\alpha)$ and $\delta_\beta = (p_\beta - p^l_\beta)/(p^r_\beta - p^l_\beta)$.

Finally, the weighted sum is conducted to calculate output values Φ^c of a 2D LUT transform like

$$\begin{aligned}\Phi^c_{(x,y)} &= w^c_{rg} \cdot \phi^c_{rg,(x,y)} \\ &+ w^c_{rb} \cdot \phi^c_{rb,(x,y)} \\ &+ w^c_{rb} \cdot \phi^c_{rb,(x,y)} + b^c,\end{aligned} \tag{5}$$

where w^c_{rg}, w^c_{rb}, w^c_{gb}, and b^c are generated weights by LUT weight generator $H_{tw}(\cdot)$ in Sec. 1.

3.2. 2D Slicing

As can be seen in Fig. 2b, the 2D slicing operation $Slicing^{c'_k}_{2D}(X, G^{2D})$ is similar to the LUT transform since both operations are based on bilinear interpolation. First, we find the query points based on the spatial coordinate (x, y) and the corresponding color value on each channel of the image, which can be denoted as

$$\begin{aligned}P_{xy} &= (p_x, p_y) = (x', y'), \\ P_{xc} &= (p_x, p_c) = (x', X_{(c'_{k'},x,y)}), \\ P_{yc} &= (p_y, p_c) = (y', X_{(c'_{k'},x,y)}),\end{aligned} \tag{6}$$

where $x' = x/(W-1)$, $y' = y/(H-1)$ and $k' = mod(k, 3)$.

Second, the bilinear interpolation is also carried out on 2D bilateral grids $g^{c'_k}_{\alpha\beta}$ for retrieved values $\psi^{c'_k}_{\alpha\beta}$ with $\alpha\beta \in \{xy, xc, yc\}$. The neighboring vertices and adjacent points of each query point for slicing can be defined as

$$\begin{aligned}p^l_\alpha &= \lfloor p_\alpha \cdot (D_s - 1) \rfloor / (D_s - 1), \\ p^r_\alpha &= p^l_\alpha + 1/(D_s - 1), \\ p^l_\beta &= \lfloor p_\beta \cdot (D_s - 1) \rfloor / (D_s - 1), \\ p^r_\beta &= p^l_\beta + 1/(D_s - 1),\end{aligned} \tag{7}$$

$$\begin{aligned}V^{c'_k,00}_{\alpha\beta} &= g^{c'_k}_{\alpha\beta}(p^l_\alpha, p^l_\beta), \\ V^{c'_k,10}_{\alpha\beta} &= g^{c'_k}_{\alpha\beta}(p^r_\alpha, p^l_\beta), \\ V^{c'_k,01}_{\alpha\beta} &= g^{c'_k}_{\alpha\beta}(p^l_\alpha, p^r_\beta), \\ V^{c'_k,11}_{\alpha\beta} &= g^{c'_k}_{\alpha\beta}(p^r_\alpha, p^r_\beta).\end{aligned} \tag{8}$$

The retrieved values by slicing can be formulated as

$$\begin{aligned}\psi^{c'_k}_{\alpha\beta,(x,y)} &= (1 - \delta_\alpha) \cdot (1 - \delta_\beta) \cdot V^{c'_k,00}_{\alpha\beta} \\ &+ \delta_\alpha \cdot (1 - \delta_\beta) \cdot V^{c'_k,10}_{\alpha\beta} \\ &+ (1 - \delta_\alpha) \cdot \delta_\beta \cdot V^{c'_k,01}_{\alpha\beta} \\ &+ \delta_\alpha \cdot \delta_\beta \cdot V^{c'_k,11}_{\alpha\beta}.\end{aligned} \tag{9}$$

Finally, output values $\Psi^{c'_k}$ of 2D slicing can be calculated through a weighted sum with generated weights by the bilateral grid weight generator $H_{sw}(\cdot)$ in Sec. 1. The weighted sum can be denoted as

$$\begin{aligned}\Psi^{c'_k}_{(x,y)} &= w^{c'_k}_{xy} \cdot \psi^{c'_k}_{xy,(x,y)} \\ &+ w^{c'_k}_{xc} \cdot \psi^{c'_k}_{xc,(x,y)} \\ &+ w^{c'_k}_{yc} \cdot \psi^{c'_k}_{yc,(x,y)} + b^{c'_k}.\end{aligned} \tag{10}$$

3.3. Slicing and LUT Transform

Using notations in previous sections, the cache-effective slicing and LUT transform in the main paper can be rewritten as

$$\begin{aligned}Y_{(r,x,y)} &= \Phi^r_{(x,y)} + \sum_{k=0}^{K/3-1} \Psi^{3k}_{(x,y)}, \\ Y_{(g,x,y)} &= \Phi^g_{(x,y)} + \sum_{k=0}^{K/3-1} \Psi^{1+3k}_{(x,y)}, \\ Y_{(b,x,y)} &= \Phi^b_{(x,y)} + \sum_{k=0}^{K/3-1} \Psi^{2+3k}_{(x,y)}.\end{aligned} \tag{11}$$

4. Additional Quantitative and Qualitative Comparisons

In this section, we provide additional quantitative comparisons on HDRTV1K dataset [2]. Additional qualitative comparisons are conducted on FiveK [1], PPR10K [7], and HDRTV1K [2].

4.1. Additional Dataset

The HDRTV1K is a dataset for the SDRTV-to-HDRTV task, which converts SDR contents to their HDRTV version. This dataset comprises captured images from 22 HDR10 videos and their corresponding SDR versions. All HDR10 videos are encoded using PQ-OETF and the rec.2020 gamut. The 1235 images from 18 videos are used in the training stage, and 117 images from 4 videos are used in the testing stage.

Table 6. Quantitative comparisons on HDRTV1K [2]. The best and second-best results are in red and blue, respectively.

Method	PSNR	SSIM	ΔE_{ITP}	HDR-VPD3	Runtime(ms)
HDRNet [3]	35.73	0.9664	11.52	8.462	56.07
CSRNet [4]	35.04	0.9625	14.28	8.400	77.1
3DLUT [12]	36.06	0.9609	10.73	8.353	1.04
AdaInt [10]	36.22	0.9658	10.89	8.423	1.59
SABLUT [6]	36.41	0.9657	10.28	8.460	3.64
HDRTVNet [2]	36.88	0.9655	9.78	8.464	70.01
Ours	36.74	0.9663	9.99	8.500	1.38

4.2. Additional Quantitative Comparisons

We compare our method with other SOTA real-time methods [3, 4, 6, 10, 12] and HDRTVNet [2] on the HDRTV1K

[2]. HDRTVNet is a method for the SDRTV-to-HDRTV task, which is introduced together with the HDRTV1K dataset in their paper [2]. HDRTVNet is set to the fastest configuration to compare the real-time performance, which only uses the adaptive global color mapping.

We measure PSNR, SSIM, ΔE_{ITP} [5], and HDR-VPD3 [8]. The ΔE_{ITP} is the color difference on the ICtCp space and is designed for HDRTV. HDR-VDP3 is an improved version of HDR-VDP2 that supports the rec.2020 gamut. We measure these metrics using codes in the official repository of HDRNet. As the HDRTV1K dataset was captured from video sequences, this experiment can offer insights into the performance on video.

As can be seen in Tab. 6, our method shows suitable performance and inference time on real-time video processing. Although HDRTVNet has the best score on PSNR and ΔE_{ITP}, it fails to achieve real-time performance under the fastest configuration. Our model delivers real-time processing with a minor performance drop: 0.11 dB on PSNR, 0.0001 on SSIM, and 0.11 on ΔE_{ITP}.

4.3. Additional Qualitative Comparisons

We provide additional qualitative results in Fig. 3, Fig. 4 and Fig. 5.

References

[1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011. 1, 4, 6

[2] Xiangyu Chen, Zhengwen Zhang, Jimmy S Ren, Lynhoo Tian, Yu Qiao, and Chao Dong. A new journey from sdrtv to hdrtv. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4500–4509, 2021. 4, 5, 8

[3] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 4

[4] Jingwen He, Yihao Liu, Yu Qiao, and Chao Dong. Conditional sequential modulation for efficient global image retouching. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 679–695. Springer, 2020. 4

[5] ITU-R. Objective metric for the assessment of the potential visibility of colour differences in television. ITU-R Rec BT.2124-0, 2019. 5

[6] Wontae Kim and Nam Ik Cho. Image-adaptive 3d lookup tables for real-time image enhancement with bilateral grids. In *European Conference on Computer Vision*, pages 91–108, 2024. 1, 4

[7] Jie Liang, Hui Zeng, Miaomiao Cui, Xuansong Xie, and Lei Zhang. Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 653–661, 2021. 1, 4, 7

[8] Rafat Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on graphics (TOG)*, 30(4):1–14, 2011. 5

[9] Tao Wang, Yong Li, Jingyang Peng, Yipeng Ma, Xian Wang, Fenglong Song, and Youliang Yan. Real-time image enhancer via learnable spatial-aware 3d lookup tables. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2471–2480, 2021. 1

[10] Canqian Yang, Meiguang Jin, Xu Jia, Yi Xu, and Ying Chen. Adaint: Learning adaptive intervals for 3d lookup tables on real-time image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17522–17531, 2022. 1, 4

[11] Canqian Yang, Meiguang Jin, Yi Xu, Rui Zhang, Ying Chen, and Huaida Liu. Seplut: Separable image-adaptive lookup tables for real-time image enhancement. In *European Conference on Computer Vision (ECCV)*, pages 201–217. Springer, 2022. 1

[12] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44 (4):2058–2073, 2022. 1, 4

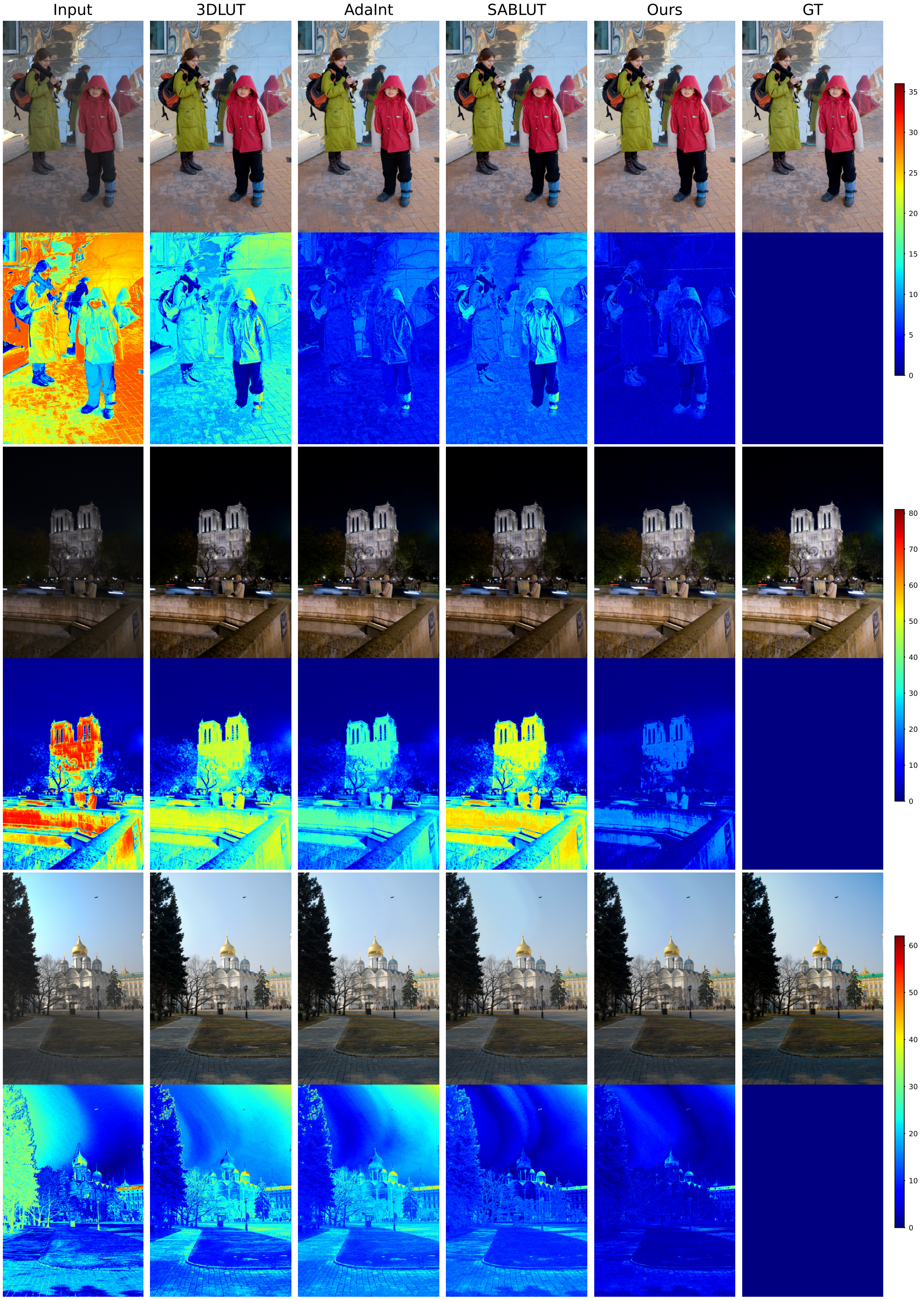


Figure 3. Qualitative comparisons for photo retouch task on the FiveK dataset [1]. The error maps at the bottom of each picture present differences with ground truth. Each color on error map indicates the degree of error based on the corresponding color bars on the right.

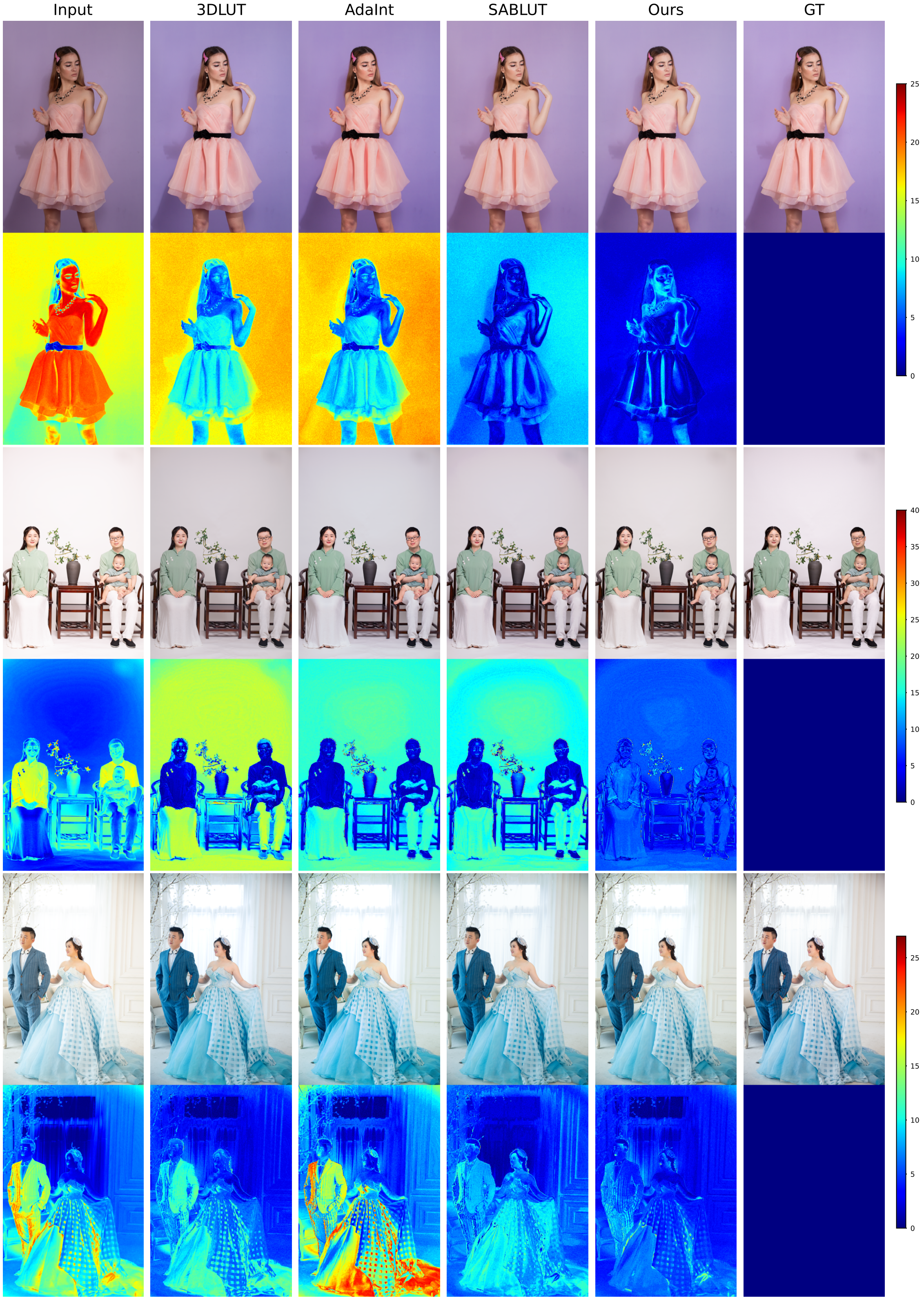


Figure 4. Qualitative comparisons for photo retouch task on the PPR10K dataset [7]. The error maps at the bottom of each picture present differences with ground truth. Each color on error map indicates the degree of error based on the corresponding color bars on the right.

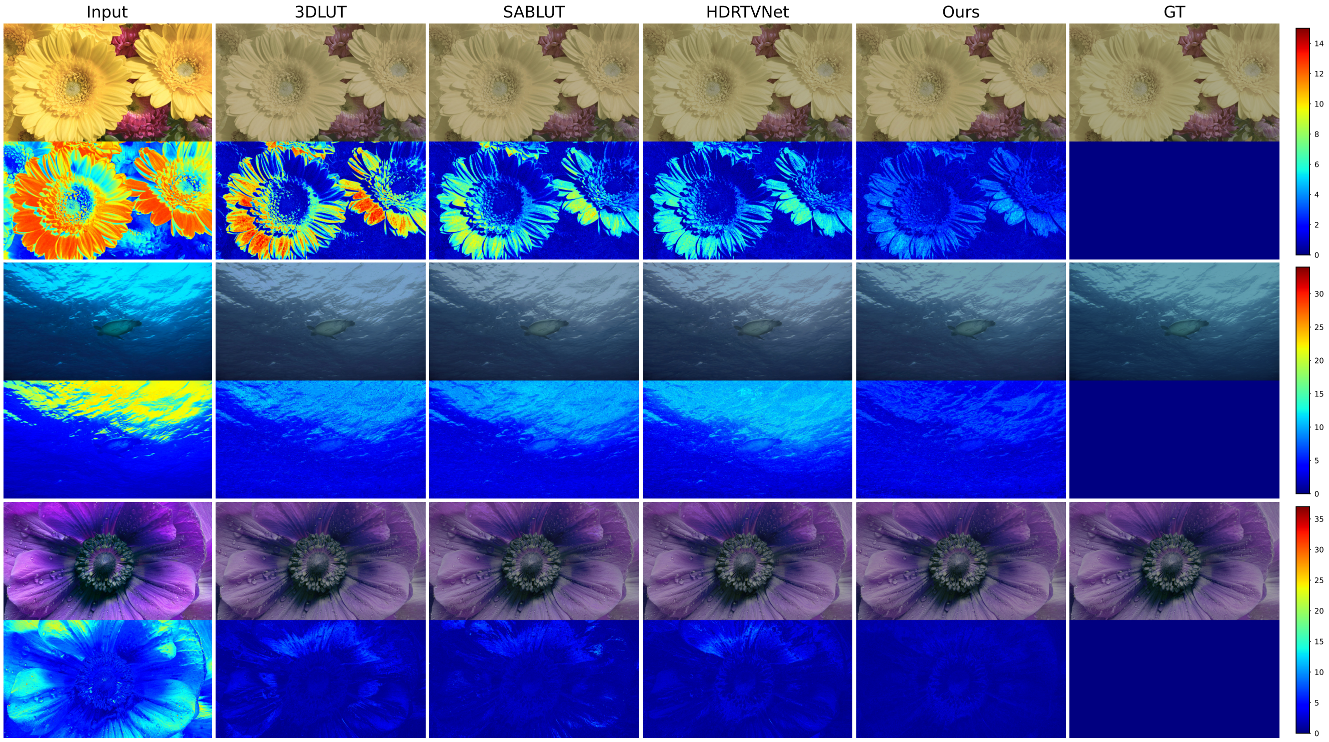


Figure 5. Qualitative comparisons for SDRTV-to-HDRTV task on the HDRTV1K dataset [2]. The error maps at the bottom of each picture present differences with ground truth. Each color on error map indicates the degree of error based on the corresponding color bars on the right.