Deep Video Inverse Tone Mapping Based on Temporal Clues Supplementary Material

1 Overview

In this supplementary material, additional explanations and analyses have been provided. Firstly, the details of the testing dataset are given in Section 2, and the details of the synthetic training dataset in Section 3. Secondly, details of the proposed network architectures are shown in Section 4. Thirdly, additional visual video results are attached into the supplementary material.

2 Testing dataset

At first we use the inverse camera response curve with gamma 2.2 to map the LDR frames of REDS-val [7] into linear domain and amplify them with scale factor 8 to form the corresponding HDR version REDS-val-hdr. As specified in the manuscript, there is no corresponding LDR version for the testing real-world HDR datasets HDM-HDRv [1], LiU-HDRv [3], and MPI-HDRv [2], we generate the LDR videos from them by simulating the camera imaging pipeline. Specifically, for each sequence of these dataset, we simulate the camera auto exposure process to set a exposure value for them. Unfortunately, the auto exposure algorithms are usually top of the secret for the camera companies and we cannot get an applicable auto exposure method from the Internet. Therefore, we try to form the camera auto exposure process by ourselves. We first define the luminance of the current frame L_{frame} as the weighting of the luminance of the region of interest L_{roi} and the overall average luminance L_{avg}:

$$L_{frame} = \alpha L_{roi} + (1 - \alpha) L_{avg}, \tag{1}$$

where α is set to 0.7 and the region of interest is defined as in Fig. 1. Then we search an appropriate exposure value T for each frame to make the L_{frame} of preliminary generated LDR frame I to be in [0.395,0.405]. The process of formulating LDR frames is:

$$I = f(clip(H \cdot T)), \tag{2}$$

where H is the HDR frame and f is the camera response curve with gamma 1/2.2. Finally, we smooth the exposure value T in temporal domain to avoid flicker:

$$T_i = \beta T_i + (1 - \beta) T_{i-1}, \tag{3}$$

where i is the temporal index and β is set to 0.8. Finally, we use T_i to generate the LDR sequence as in Eq. (2).

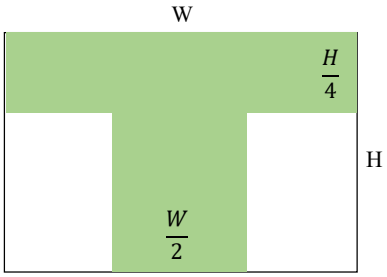


Figure 1: The illustration of the region of interest.

3 Examples of Synthetic Dataset

As description in Sec 3.5 of the manuscript, we derive our training dataset from both HDR image datasets and LDR video datasets. By this way, we can simultaneously exploit the video characteristics of LDR videos and the HDR characteristics of HDR images. Fig. 2 shows examples of the synthetic dataset. First, we apply a random perspective transformation on HDR images to simulate camera motion to obtain hdr videos as shown in 1st row of Fig.2. And as the way we obtain REDS-val-hdr, we get corresponding HDR version REDS-hdr of training set of REDS[7], which are shown in 3nd row of Fig 2. Then, we derive LDR videos from these hdr videos as shown in 2nd and 4th rows of Fig 2, where we set a high exposure duration T for the current frame as input and random T for others frames in these sequences.

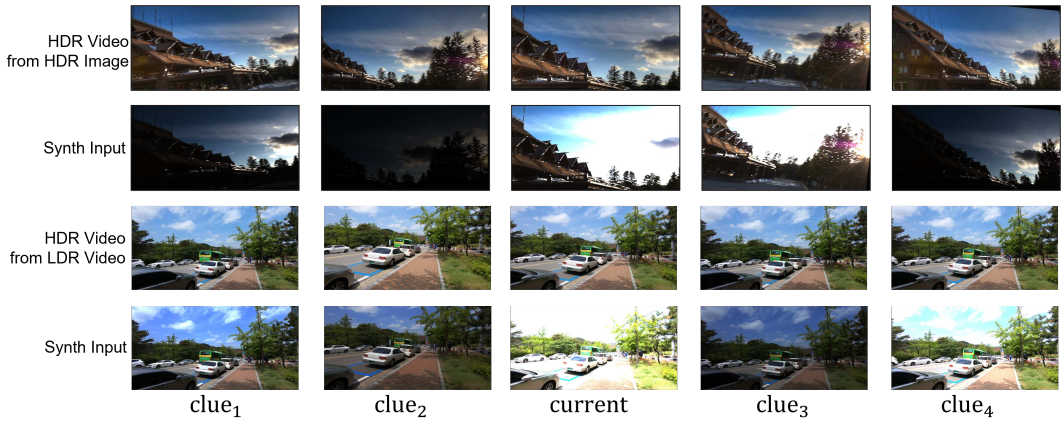


Figure 2: Examples of the Synthetic Dataset.

4 Details of the proposed models

The detailed architectures of the proposed components are shown in Fig.3. The correction convolutional block and fusion convolutional block used in the proposed Local Feature Alignment Block are shown in Fig.3 (a) and Fig.3 (b) separately. The swim transformer block used in the proposed Global Feature Aggregation Block is shown in Fig.3 (c) and the ExpandNet [6] used in the proposed Feature and Clue Propagation Module is shown in Fig.3 (d). We adopt the swim transformer block used in [5] [4]. As specified in [4], given an input of size $H \times W \times C$, swim transformer block first reshapes the input to a $\frac{HW}{M^2} \times M^2 \times C$ feature by partitioning the input into non-overlapping $M \times M$ local windows, where $\frac{HW}{M^2}$ is the total number of windows. Then, it computes the standard self-attention separately for each window. For a local window feature X, the query, key and value matrices Q, K and V are computed as:

$$Q = XP_Q, K = XP_K, V = XP_V, \tag{4}$$

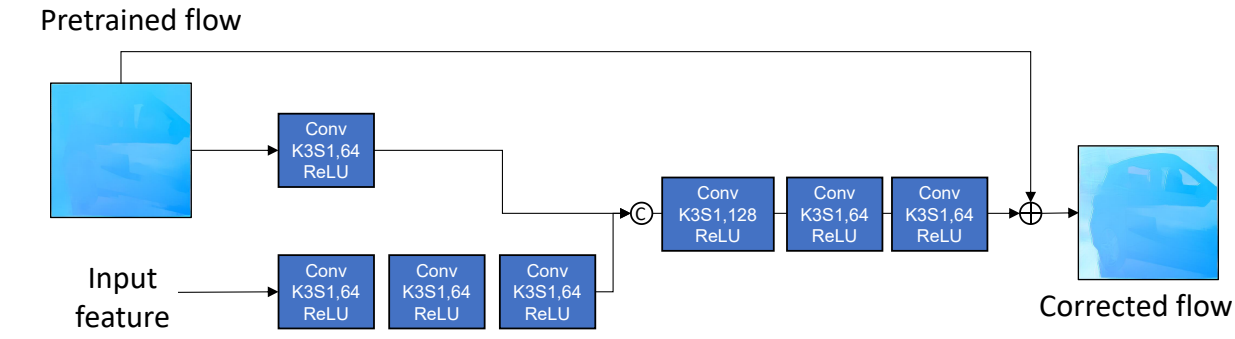
where P_Q, P_K, and P_V are projection matrices that are shared across different windows. Generally, we have $Q, K, V \in \mathbb{R}^{M^2 \times d}$. The attention matrix is thus computed by the self-attention mechanism in a local window as:

$$Attention Q, K, V = SoftMax(QK^T/\sqrt{d} + B)V, \tag{5}$$

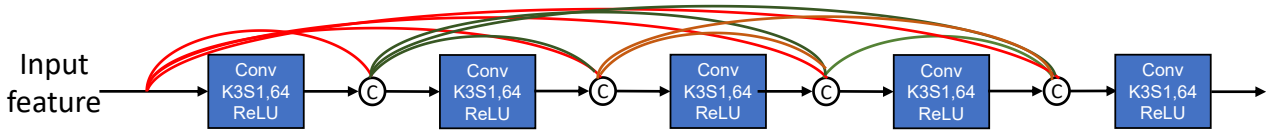
where B is the learnable relative positional encoding. In practice, following [4], we perform the attention function for h times in parallel and concatenate the results for multihead self-attention (MSA). Next, a multi-layer perceptron (MLP) that has two fullyconnected layers with GELU non-linearity between them is used for further feature transformations. The LayerNorm (LN) layer is added before both MSA and MLP, and the residual connection is employed for both modules. The whole process is formulated as:

$$X = MSA(LN(X)) + X, X = MLP(LN(X)) + X. \tag{6}$$

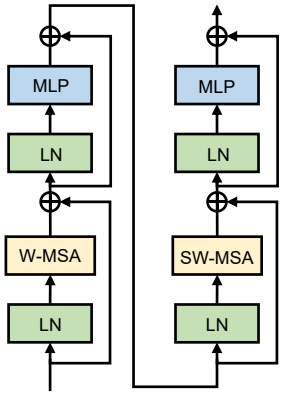
Furthermore, regular and shifted window partitioning are used alternately to enable cross-window connections [5], where shifted window partitioning means shifting the feature by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels before partitioning.



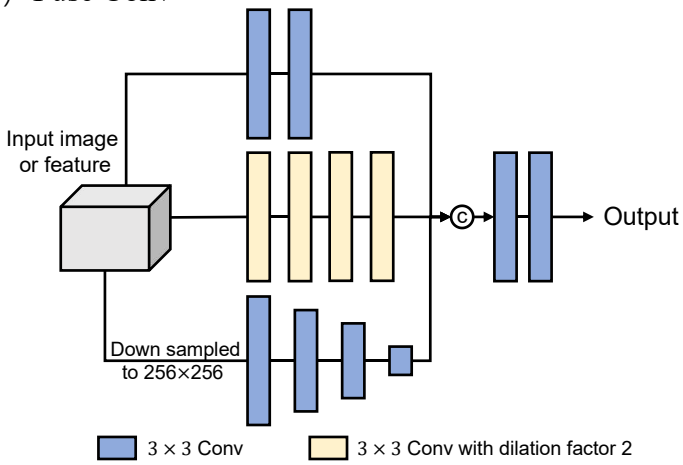
(a) Correction Conv



(b) Fuse Conv



(c) Swin Transformer Block



(d) ExpandNet

Figure 3: The details of the proposed component.

References

[1] Jan Froehlich, Stefan Grandinetti, Bernd Eberhardt, Simon Walter, Andreas Schilling, and Harald Brendel. Creating cinematic wide gamut hdr-video for the evaluation of tone mapping operators and hdr-displays. In *Digital photography X*, volume 9023, pages 279–288. SPIE, 2014.

[2] Vlastimil Havran, Miloslaw Smyk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Interactive system for dynamic scene lighting using captured video environment maps. In *Rendering Techniques*, pages 31–42, 2005.

[3] Joel Kronander, Stefan Gustavson, Gerhard Bonnet, Anders Ynnerman, and Jonas Unger. A unified framework for multi-sensor hdr video reconstruction. *Signal Processing: Image Communication*, 29(2):203–215, 2014.

[4] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.

[5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[6] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debattista. Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. In *Computer Graphics Forum*, volume 37, pages 37–49. Wiley Online Library, 2018.

[7] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, June 2019.