Deep Video Harmonization with Color Mapping Consistency

Xinyuan Lu[*] , **Shengyuan Huang**[*] , **Li Niu**[†] , **Wenyan Cong** , **Liqing Zhang**[†]
MoE Key Lab of Artificial Intelligence, Department of Computer Science and Engineering
Shanghai Jiaotong University, Shanghai, China
{lxy9807, huangshengyuan, ustcnewly, plcwyam17320}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract

Video harmonization aims to adjust the foreground of a composite video to make it compatible with the background. So far, video harmonization has only received limited attention and there is no public dataset for video harmonization. In this work, we construct a new video harmonization dataset HYouTube by adjusting the foreground of real videos to create synthetic composite videos. Moreover, we consider the temporal consistency in video harmonization task. Unlike previous works which establish the spatial correspondence, we design a novel framework based on the assumption of color mapping consistency, which leverages the color mapping of neighboring frames to refine the current frame. Extensive experiments on our HYouTube dataset prove the effectiveness of our proposed framework. Our dataset and code are available at https://github.com/bcmi/Video-Harmonization-Dataset-HYouTube.

1 Introduction

Given two different videos, video composition aims to generate a composite video by combining the foreground of one video with the background of another video. However, composite videos are usually not realistic enough due to the appearance (*e.g.*, illumination, color) incompatibility between foreground and background, which is caused by distinctive capture conditions (*e.g.*, season, weather, time of the day) of foreground and background [Cong *et al.*, 2020; Cong *et al.*, 2021]. To address this issue, video harmonization [Haozhi *et al.*, 2019] has been proposed to adjust the foreground appearance to make it compatible with the background, resulting in a more realistic composite video.

As a closely related task, image harmonization has attracted growing research interest. Recently, several deep learning based image harmonization methods [Cong *et al.*, 2020; Cun and Pun, 2020; Guo *et al.*, 2021; Cong *et al.*, 2021; Konstantin *et al.*, 2021; Ling *et al.*, 2021] have been proposed. They changed the foreground style to be harmonious with the background using deep learning techniques. However, directly applying them to video harmonization by harmonizing each frame separately will cause flickering artifacts [Haozhi *et al.*, 2019]. So [Haozhi *et al.*, 2019] considered the temporal consistency between adjacent frames and proposed an end-to-end network to harmonize the composite frames.

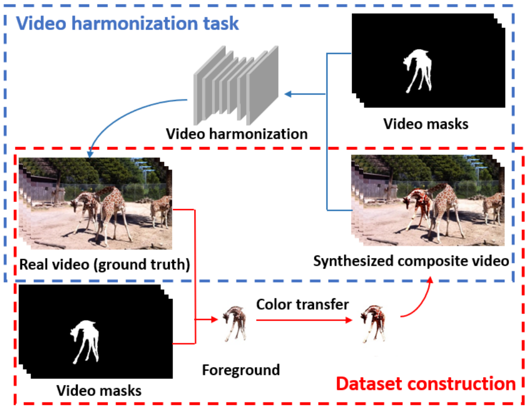


Figure 1: Illustration of video harmonization task (blue arrows) and dataset construction process (red arrows).

Training deep video harmonization network requires abundant pairs of composite videos and their ground-truth harmonized videos, but manually editing composite videos to obtain their harmonized videos is tedious and expensive. Therefore, [Haozhi *et al.*, 2019] adopted an inverse approach by applying the traditional color transfer method [Reinhard *et al.*, 2001] to the foreground of the real image to make it incompatible with the background, leading to the synthetic composite image. Then, they applied affine transformation to foreground and background to simulate the motion between adjacent frames, through which synthetic composite video (*resp.*, ground-truth video) are created based on synthetic composite image (*resp.*, real image). Nevertheless, there is a huge gap between the simulated movement and the complex movement in realistic videos. Moreover, their dataset is not publicly available. Different from [Haozhi *et al.*, 2019], we create synthetic composite videos based on real video without sacrificing realistic motion. Specifically, we apply color transfer based on lookup table (LUT) to the foregrounds of all frames and the details will be introduced in Section 3. We construct

[*]Both authors contributed equally to this research.
[†]Corresponding Authors.

our video harmonization dataset named HYouTube based on YouTube-VOS 2018 [Xu *et al.*, 2018], leading to 3194 pairs of synthetic composite videos and real videos.

To alleviate flickering artifacts, we also propose a novel video harmonization framework considering temporal coherence. Previous video-related methods usually seek for spatial correspondence, *i.e.*, the relative movement of pixels/regions between adjacent frames, using optical flow or attention mechanism. However, it is time-consuming and challenging to establish the spatial correspondence accurately [Liu *et al.*, 2020]. To escape from establishing spatial correspondence, we use color mapping consistency to enhance temporal coherence. In real videos, the color statistics between two adjacent frames should be close for both foreground and background regions. Therefore, we assume that the color transformations (mappings) to harmonize two adjacent composite frames should also be close.

Based on the assumption of color mapping consistency, we propose a framework consisting of an image harmonization network and a refinement module. Specifically, we first employ existing image harmonization network to harmonize composite frames. Then, for each composite frame, we can summarize the color transformation of its neighboring frames and compact it in a lookup table (LUT). Next, we apply the summarized LUT to this composite frame to obtain the LUT result. Finally, we feed the image harmonization result and the LUT result into a light-weighted refinement module to get our final result, during which the LUT result ensures color mapping consistency and helps improve temporal coherence. We name our framework as **CO**lor mapping **CO**nsistency **Net**work (CO$_2$Net). Our major contributions can be summarized as follows:

- We contribute the first public video harmonization dataset named HYouTube.
- To ensure temporal coherence, we propose a novel video harmonization framework named CO$_2$Net based on the assumption of color mapping consistency.

2 Related Work

2.1 Image Harmonization

Image harmonization aims to adjust the foreground appearance to match the background appearance. Recently, deep learning based image harmonization methods [Yi-Hsuan *et al.*, 2017; Cong *et al.*, 2020; Cong *et al.*, 2022] mainly concentrated on learning transformation based on image-to-image translation [Phillip *et al.*, 2016]. [Cong *et al.*, 2020] proposed a domain verification discriminator to help translate the foreground to the same domain of background. [Cong *et al.*, 2021] and [Ling *et al.*, 2021] utilized the background information as guidance to translate the foreground. The works in [Cun and Pun, 2020; Hao *et al.*, 2020] leveraged various attention mechanisms to improve the performance of harmonization networks. [Guo *et al.*, 2021] decomposed the composite image into reflectance and illumination according to Retinex theory, and harmonized the illumination map.

2.2 Video Harmonization

When applying the image harmonization methods to harmonize each composite frame individually, there will be notable flickering artifacts. To solve this issue, video harmonization method [Haozhi *et al.*, 2019] harmonized each composite frame and leveraged optical flow to make the aligned harmonized results of adjacent frames consistent. Different from this method, we leverage color mapping consistency instead of spatial correspondence to ensure the temporal coherence of a harmonized video.

2.3 Image and Video Harmonization Datasets

Since it is very difficult to obtain pairs of composite images and their ground-truth harmonized images, pioneering works [Yi-Hsuan *et al.*, 2017; Cong *et al.*, 2020] adopted an inverse approach by translating real images to synthetic composite images. The first public image harmonization dataset is iHarmony4 [Cong *et al.*, 2020], which consists of four subdatasets: HCOCO, HFlickr, HAdobe5k, Hday2night.

For video harmonization dataset, similar to [Cong *et al.*, 2020], [Haozhi *et al.*, 2019] first converted real images to synthetic composite images using color transfer method. Then, affine transformation is applied to the foregrounds/backgrounds in real images and synthetic composite images synchronously to simulate the movement between adjacent frames, leading to pairs of synthetic composite videos and harmonized videos. However, the simulated movement is dramatically different from realistic motion. In contrast, we construct a video harmonization dataset by translating real videos to synthetic composite videos.

3 Dataset Construction

We construct our dataset HYouTube based on the large-scale video object segmentation dataset YouTube-VOS 2018 [Xu *et al.*, 2018]. Given real videos with object masks, we first select the videos which meet our requirements and then adjust their foregrounds to produce synthetic composite videos.

3.1 Real Video Selection

YouTube-VOS 2018 [Xu *et al.*, 2018] contains 4453 YouTube video clips and each video clip is annotated with the object masks for one or multiple objects. Each second has 6 frames with mask annotations and we only utilize these annotated frames. Then, for each annotated foreground object in each video clip, if there exist more than 20 consecutive frames containing this foreground object, we save the first 20 consecutive frames with the corresponding 20 foreground masks as one video sample. After that, we remove the video samples with foreground ratio (the area of foreground over the area of the whole frame) smaller than 1% to ensure that the foreground area is in a reasonable range. After the above filtering steps, there are 3194 video samples left.

3.2 Composite Video Generation

Based on real video samples, we adjust the appearance of their foregrounds to make them incompatible with backgrounds, producing synthetic composite videos. We have tried different color transfer methods and 3D color lookup table (LUT) to adjust the foreground appearance. The color transfer methods [Kalyan *et al.*, 2010; Su *et al.*, 2012; Lalonde and Efros, 2007; Zhu *et al.*, 2015] need a reference image and adjust the source image appearance based on the reference image appearance, while LUT [Mese Murat, 2001; Bo *et al.*, 2011] is a simple array indexing operation to realize color mapping. The details of LUT will be introduced in Section 4.2. We observe that applying color transfer methods requires carefully picking reference images, otherwise the transferred foreground may have obvious artifacts or look unrealistic. Thus, we employ LUT to adjust the foreground appearance for convenience. Since one LUT corresponds to one type of color transfer, we can ensure the diversity of the composite videos by applying different LUTs to video samples. We collect more than 400 LUTs from the Internet and select 100 candidate LUTs among them with the largest mutual difference (see Supplementary).

The process of generating composite video samples is illustrated in Figure 1. Given a video sample, we first select an LUT from 100 candidate LUTs randomly to transfer the foreground of each frame. The transferred foregrounds and the original backgrounds form the composite frames, and the composite frames form the composite video samples. Following [Cong *et al.*, 2020], we set some rules to filter out unqualified composite video samples, which can be found in Supplementary. We name our constructed video harmonization dataset as HYouTube, which includes 3194 pairs of synthetic composite video samples and real video samples. Each video sample contains 20 consecutive frames with the foreground mask for each frame.

4 Our Method

4.1 Framework Overview

We propose a video harmonization framework consisting of an image harmonization network G (*e.g.*, [Ling *et al.*, 2021; Konstantin *et al.*, 2021]) and a refinement module R as shown in Figure 2. Given an input composite video, each frame $\mathbf{I}_i$ concatenated with its foreground mask $\mathbf{M}_i$ goes through the image harmonization network to obtain the image harmonization result $\tilde{\mathbf{I}}^h_i$.

Then, the refinement module takes the neighboring frames into consideration and utilizes color mapping consistency to enhance the harmonized result. Specifically, given the current frame $\mathbf{I}_i$, we calculate the color mapping based on its neighboring frames $\{\mathbf{I}_{i-T}, \ldots, \mathbf{I}_{i-1}, \mathbf{I}_{i+1}, \ldots, \mathbf{I}_{i+T}\}$ and image harmonization results $\{\tilde{\mathbf{I}}^h_{i-T}, \ldots, \tilde{\mathbf{I}}^h_{i-1}, \tilde{\mathbf{I}}^h_{i+1}, \ldots, \tilde{\mathbf{I}}^h_{i+T}\}$, where T is the number of neighbors on one side. We calculate the color mapping $f : \mathbf{I} \mapsto \tilde{\mathbf{I}}^h$ in the form of lookup table (LUT), which will be detailed in Section 4.2. We assume that the color mapping of the current frame should be consistent with its neighboring frames. Thus, we apply the color mapping f calculated based on neighboring frames to the current frame $\mathbf{I}_i$ to obtain the LUT result $\tilde{\mathbf{I}}^l_i = f(\mathbf{I}_i)$. Finally, we feed LUT result $\tilde{\mathbf{I}}^l_i$ and image harmonization result $\tilde{\mathbf{I}}^h_i$ into a light-weighted refinement module to achieve the refined result $\tilde{\mathbf{I}}^r_i$.

4.2 Color Mapping based on Lookup Table

Lookup table (LUT) records the input color and the corresponding output color, so one LUT corresponds to one color mapping function f. LUT has been applied in a variety of computer vision tasks like image enhancement [Bo *et al.*, 2011; Fischl and Schwartz, 1999] and image denoising [Michael *et al.*, 2006; Shu *et al.*, 2015]. As shown in Figure 2(b), an LUT is a 3D lattice in the RGB space and each dimension corresponds to one color channel (*e.g.*, red). LUT consists of $V = (B + 1)^3$ entries by uniformly discretizing the RGB color space, where B is the number of bins in each dimension. Each entry v in the LUT has an indexing color $\mathbf{c}'_v = (r'_v, g'_v, b'_v)$ and its corresponding output color $\tilde{\mathbf{c}}'_v = (\tilde{r}'_v, \tilde{g}'_v, \tilde{b}'_v)$. The color transformation process based on LUT has two steps: *look up* and *trilinear interpolation*. Specifically, given a color value, we first look up its eight nearest entries in the LUT, and then interpolate its transformed value based on eight nearest entries via trilinear interpolation.

As introduced in Section 4.1, given the current frame $\mathbf{I}_i$, we tend to calculate the color mapping f (an LUT) based on its neighboring frames. Then, we apply the color mapping f to the current frame $\mathbf{I}_i$ to obtain the LUT result. Next, we will elaborate on these two stages one by one.

Calculating LUT

For ease of description, given the current frame $\mathbf{I}_i$, we collect the foreground pixels of its neighboring frames $\{\mathbf{I}_{i-T}, \ldots, \mathbf{I}_{i-1}, \mathbf{I}_{i+1}, \ldots, \mathbf{I}_{i+T}\}$ as an array of pixels $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\}$, in which N is the total number of pixels. With their image harmonization results, we can obtain the harmonized pixels $\tilde{\mathcal{C}} = \{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \ldots, \tilde{\mathbf{c}}_N\}$. Our goal is learning an LUT (color mapping f) which can translate $\mathcal{C}$ to $\tilde{\mathcal{C}}$. However, it is time-consuming and inconvenient to directly solve the optimization problem $\min_f \frac{1}{3N} \sum_{n=1}^{N} \|f(\mathbf{c}_n) - \tilde{\mathbf{c}}_n\|^2$. Hence, we design a heuristic approach which is effective in practice and much more efficient than direct optimization (see Section 5.5). The intuitive idea of our heuristic approach is as follows: 1) for each $\mathbf{c}_n$, look up its nearest entries in the LUT and assign the harmonized value $\tilde{\mathbf{c}}_n$ to these entries; 2) for each entry in the LUT, aggregate the assigned values as the output color of this entry.

For each pixel $\mathbf{c}_n$, we calculate its similarity with each indexing color $\mathbf{c}'_v$ in the LUT. Given a pixel value $\mathbf{c}_n = (r_n, g_n, b_n)$ and an indexing color $\mathbf{c}'_v = (r'_v, g'_v, b'_v)$, the similarity between $\mathbf{c}_n$ and $\mathbf{c}'_v$ is calculated as

$$w_{\mathbf{c}_n, \mathbf{c}'_v} = \prod_{z \in \{r,g,b\}} max(0, 1 - \frac{|z_n - z'_v|}{d}), \qquad (1)$$

in which $d = 256/B$ is bin size. According to (1), it can be seen that $\mathbf{c}_n$ only has non-zero similarities with its eight nearest entries and the similarities share the same form as the interpolation coefficients in trilinear interpolation. We assign the harmonized pixel $\tilde{\mathbf{c}}_n$ to the eight nearest entries (*e.g.*, $\mathbf{c}'_v$) of $\mathbf{c}_n$ with different weights (*e.g.*, $w_{\mathbf{c}_n, \mathbf{c}'_v}$). After traversing all $\mathbf{c}_n$, we can calculate the output value in each entry in the

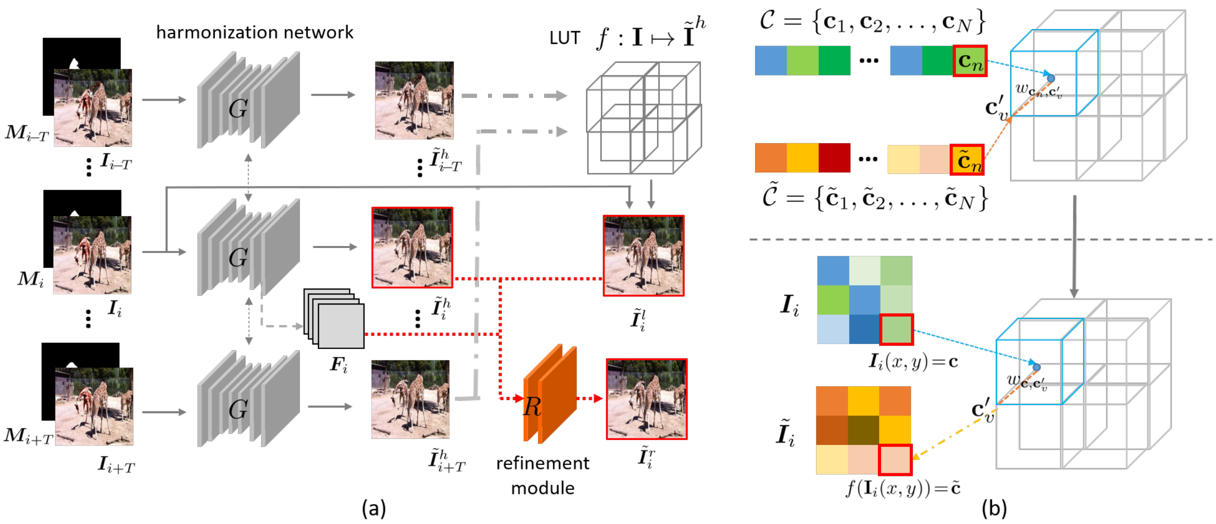

harmonization network
LUT $f : \mathbf{I} \mapsto \tilde{\mathbf{I}}^h$
M_{i-T}
I_{i-T}
G
$\tilde{I}^h_{i-T}$
M_i
I_i
F_i
$\tilde{I}^h_i$
$\tilde{I}^l_i$
M_{i+T}
I_{i+T}
$\tilde{I}^h_{i+T}$
R
refinement module
$\tilde{I}^r_i$
(a)
$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\}$
$\mathbf{c}_n$
$w_{\mathbf{c}_n,\mathbf{c}'_v}$
$\mathbf{c}'_v$
$\tilde{\mathbf{c}}_n$
$\tilde{\mathcal{C}} = \{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \ldots, \tilde{\mathbf{c}}_N\}$
$\mathbf{I}_i$
$\mathbf{I}_i(x,y)=\mathbf{c}$
$w_{\mathbf{c},\mathbf{c}'_v}$
$\mathbf{c}'_v$
$\tilde{\mathbf{I}}_i$
$f(\mathbf{I}_i(x,y))=\tilde{\mathbf{c}}$
(b)

Figure 2: The subfigure (a) illustrates the whole framework. For the current frame $\mathbf{I}_i$, we can get its image harmonization result $\tilde{\mathbf{I}}^h_i$ and LUT result $\tilde{\mathbf{I}}^l_i$, which are fed into the refinement module together with decoder feature $\mathbf{F}_i$ to produce the refined result $\tilde{\mathbf{I}}^r_i$. The subfigure (b) depicts the process of calculating LUT based on neighboring frames (top) and applying LUT to the current frame (bottom).

LUT as the weighted average of harmonized pixels:

$$\tilde{\mathbf{c}}'_v = \frac{\sum_{n=1}^{N} w_{\mathbf{c}_n,\mathbf{c}'_v} \tilde{\mathbf{c}}_n}{\sum_{n=1}^{N} w_{\mathbf{c}_n,\mathbf{c}'_v}}. \tag{2}$$

One issue is that the pixels in $\mathcal{C}$ cannot cover all the entries in the LUT, so $\sum_{n=1}^{N} w_{\mathbf{c}_n,\mathbf{c}'_v} = 0$ for some entries, which are referred to as null entries.

Applying LUT

The resultant LUT in (2) represents the color mapping function f and we can apply f to the current frame $\mathbf{I}_i$. In particular, given a foreground pixel $\mathbf{I}_i(x,y) = \mathbf{c}$, we can use trilinear interpolation based on the LUT to get its value $\tilde{\mathbf{c}}$ in the LUT result $\tilde{\mathbf{I}}^l_i$:

$$\tilde{\mathbf{I}}^l_i(x,y) = \tilde{\mathbf{c}} = f(\mathbf{c}) = \sum_{v=1}^{V} w_{\mathbf{c},\mathbf{c}'_v} \tilde{\mathbf{c}}'_v, \tag{3}$$

in which the interpolation coefficient $w_{\mathbf{c},\mathbf{c}'_v}$ has the same form as (1). If the eight nearest entries of one pixel $\mathbf{I}_i(x,y)$ contain null entries, we ignore the null entries and normalize the remaining coefficients. If the eight nearest entries are all null entries, we refer to this pixel as an invalid pixel and replace $\tilde{\mathbf{I}}^l_i(x,y)$ with $\tilde{\mathbf{I}}^h_i(x,y)$.

4.3 Light-weighted Refinement Module

Given the current frame $\mathbf{I}_i$, we can obtain its image harmonization result $\tilde{\mathbf{I}}^h_i$ and LUT result $\tilde{\mathbf{I}}^l_i$. Since $\tilde{\mathbf{I}}^h_i$ only focuses on the current frame without considering neighboring frames, $\tilde{\mathbf{I}}^l_i$ can compensate $\tilde{\mathbf{I}}^h_i$ for the temporal consistency.

We design a light-weighted refinement module to generate the refined result $\tilde{\mathbf{I}}^r_i$ based on $\tilde{\mathbf{I}}^h_i$ and $\tilde{\mathbf{I}}^l_i$. Besides, we conjecture that the last feature map $\mathbf{F}_i \in \mathbb{R}^{W \times H \times C}$ in the image harmonization network G contains rich useful knowledge for harmonization. Thus, we also feed $\mathbf{F}_i$ into the refinement module. After concatenating $\tilde{\mathbf{I}}^h_i$, $\tilde{\mathbf{I}}^l_i$, and $\mathbf{F}_i$, we have the $H \times W \times (C+6)$ input for our refinement module. To balance the efficiency and performance, in the refinement module, we only employ two convolutional layers, with each one followed by batch normalization and ELU activation.

During training, we adopt a two-step training strategy. In the first step, we train the image harmonization network G using its original loss functions, after which we can obtain $\tilde{\mathbf{I}}^h_i$ and $\tilde{\mathbf{I}}^l_i$ for each frame. In the second step, we train the refinement module R with MSE loss $\mathcal{L} = \|\tilde{\mathbf{I}}^r_i - \hat{\mathbf{I}}_i\|^2$, in which $\hat{\mathbf{I}}_i$ is the ground-truth harmonized result. We tried training the whole framework in an end-to-end manner, but it brings no further improvement while increasing the training difficulty. Therefore, we finally adopt the two-step training strategy.

5 Experiments

5.1 Dataset Statistics

We conduct experiments on our constructed HYouTube dataset, which contains 3194 pairs of synthetic composite video samples and real video samples. We split our dataset into 2558 video samples in the training set and 636 video samples in the test set, in which the video samples created by adjusting different foregrounds in the same video are not allowed to appear in both training set and test set. More dataset statistics (*e.g.*, the number of video samples per LUT) can be found in Supplementary.

5.2 Implementation Details

Our framework can accommodate an arbitrary image harmonization network G. We adopt two backbones iS^{2}AM [Konstantin *et al.*, 2021] and RainNet [Ling *et al.*, 2021] considering their simplicity and effectiveness. We set the number of

Method	fMSE↓	MSE ↓	PSNR↑	fSSIM↑	TL↓
Composite	1029.50	151.20	30.14	0.7197	2.5315
DoveNet [Cong *et al.*, 2020]	422.84	58.51	33.96	0.8238	13.8647
IIH [Guo *et al.*, 2021]	368.92	47.30	34.25	0.8391	3.1187
RainNet [Ling *et al.*, 2021]	374.06	49.05	34.61	0.8338	4.4733
iS^{2}AM [Konstantin *et al.*, 2021]	203.77	28.90	37.38	0.8817	6.4765
Huang *et al.* (RainNet)	373.17	43.94	34.63	0.8319	4.5044
Huang *et al.* (iS^{2}AM)	199.89	27.89	37.44	0.8821	6.4893
Ours (RainNet)	325.36	43.81	35.37	0.8534	4.0694
Ours (iS^{2}AM)	186.72	26.50	37.61	0.8827	5.1126

Table 1: Comparison between different harmonization methods on our HYouTube dataset. TL is short for temporal loss.

	Base		Refinement			fMSE↓	Time(s)↓	TL↓
	$\tilde{\mathbf{I}}^h$	$\tilde{\mathbf{I}}^l$	$\tilde{\mathbf{I}}^h$	$\tilde{\mathbf{I}}^l$	$\mathbf{F}$			
1	+					203.77	0.0034	6.4765
2		+				190.53	0.0038	6.2246
3	+	+	+	+		189.07	0.0045	6.1983
4	+	+	+		+	202.89	0.0042	6.4917
5	+	+		+	+	188.53	0.0046	6.0732
6	+	+	+	+	+	186.72	0.0046	5.1126

Table 2: The ablation studies of our CO$_2$Net. The first two rows only use image harmonization result $\tilde{\mathbf{I}}^h$ or LUT result $\tilde{\mathbf{I}}^l$. $\tilde{\mathbf{I}}^h$, $\tilde{\mathbf{I}}^l$, and $\mathbf{F}$ are three types of inputs for the refinement module. TL is short for temporal loss.

Method	B	T	fMSE↓	ME↓	Time(s)↓
Ours	32	1	199.78	16.74	0.000182
Ours	32	8	190.53	25.23	0.000185
Ours	128	1	205.06	10.83	0.000284
Ours	128	8	193.21	18.40	0.000288
Op	32	1	278.17	9.14	0.085
Op	32	8	205.23	17.85	0.155
Op	128	1	919.07	1.95	0.172
Op	128	8	219.15	10.03	0.268

Table 3: Comparison of two ways to calculate LUT. "Ours" is our way and "Op" is direct optimization. ME is short for mapping error.

neighboring frames $T = 8$ by default. The number of bins is set as $B = 32$, which is practically used in image processing. The effect of T and B will be discussed in Supplementary.

We conduct all experiments using Pytorch. We train our model on a single GTX 3090 GPU for 120 epochs using Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The initial learning rate is 10^{-3}. The batch size is set to 32 for training process. We resize composite frames to 256×256 during training and testing. The random seed is set as 5. When training the refinement module, for the images without adequate neighboring frames, we perform padding by replicating the first (*resp.*, last) frame at the beginning (*resp.*, end) of the video sample.

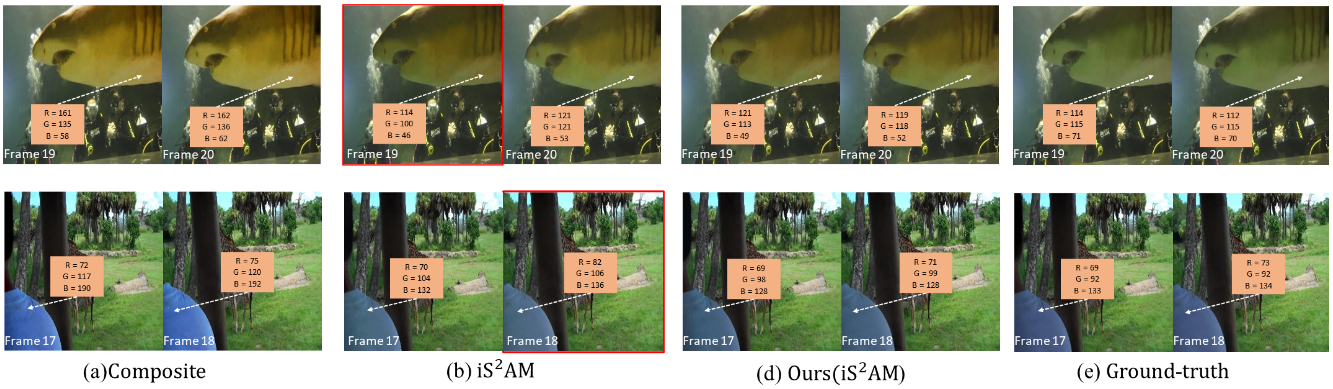
We compare different methods from two perspectives: harmonization performance and temporal consistency. For harmonization performance, we adopt MSE, fMSE, PSNR following [Cong *et al.*, 2020] and fSSIM following [Guo *et al.*, 2021], in which fMSE (*resp.*, fSSIM) means only calculating the MSE (*resp.*, SSIM) within the foreground region. For temporal consistency, we adopt Temporal Loss (TL) following [Haozhi *et al.*, 2019]. Specifically, we extract optical flows between adjacent frames and propagate the harmonized result of the previous frame to the next frame via optical flows, after which the difference between the propagated result and the original result of the next frame is calculated. More details of TL are left to Supplementary.

5.3 Comparison with Existing Methods

We compare our CO$_2$Net with two groups of baselines: image harmonization method and video harmonization method. For the first group, we compare with existing image harmonization methods iS^{2}AM [Konstantin *et al.*, 2021], RainNet [Ling *et al.*, 2021], DoveNet [Cong *et al.*, 2020], and Intrinsic Image Harmonization (IIH) [Guo *et al.*, 2021], which harmonize each individual frame separately. For the second group, the only existing video harmonization method is [Haozhi *et al.*, 2019]. For fair comparison with our model, we also use iS^{2}AM [Konstantin *et al.*, 2021] and RainNet [Ling *et al.*, 2021] as image harmonization network for [Haozhi *et al.*, 2019].

The experimental results are summarized in Table 1. Among the image harmonization methods, iS^{2}AM achieves the best results. When using iS^{2}AM or RainNet as the image harmonization network, [Haozhi *et al.*, 2019] and our method can both improve the harmonization performance and the temporal consistency, which demonstrates the effectiveness of the temporal consistency loss in [Haozhi *et al.*, 2019] and utilizing color mapping consistency in our method. Besides, our framework outperforms all baseline methods including [Haozhi *et al.*, 2019], which can be explained as follows. [Haozhi *et al.*, 2019] extracts optical flows to establish the spatial correspondences between adjacent frames, which is time-consuming and the established spatial correspondences are often inaccurate [Liu *et al.*, 2020]. In contrast, we leverage color mapping consistency to avoid establishing spatial correspondences, which is both efficient and effective.

Following [Haozhi *et al.*, 2019], we conduct user study to evaluate the harmonized videos from two aspects: realism and temporal consistency. For each test video sample, we ask 20 users to rank the results of iS^{2}AM, Huang *et al.* (iS^{2}AM), and Ours(iS^{2}AM), after which the Plackett-Luce scores are calculated for three methods. The details of user study are



(a)Composite (b) $\mathrm{iS^2AM}$ (d) Ours($\mathrm{iS^2AM}$) (e) Ground-truth

Figure 3: The qualitative comparison between $\mathrm{iS^2AM}$ and Ours ($\mathrm{iS^2AM}$) on two adjacent frames. The frames with red borders are inconsistent with neighboring frames, which cause flickering artifacts. We also show the RGB values of temporally identical pixels in two adjacent frames.

left to Supplementary.

5.4 Ablation Studies

In this section, we conduct ablation studies to analyze each component in our framework and each type of input in our refinement module in Table 2. First, we report the image harmonization result in row 1 and the LUT result in row 2. Then, we add the refinement module and explore different types of inputs. By comparing row 3-5 with row 6, we observe that all types of inputs are essential for the refinement module. By comparing the time in row 1, 2, 6, it can be seen that color mapping based on LUT only costs negligible time (time in row 2 contains the time of image harmonization network) and the refinement module is also very efficient.

5.5 LUT Calculation

As mentioned in Section 4.2, instead of directly solving the optimization problem, we design a heuristic approach to calculate the LUT, which empirically works well. We compare our approach ("Ours") with direct optimization ("Op") in different settings (*e.g.*, B, T). For direct optimization, we use gradient descent to optimize the LUT by minimizing the mapping error $\min_f \frac{1}{3N}\sum_{n=1}^{N}\|f(\mathbf{c}_n)-\tilde{\mathbf{c}}_n\|^2$. We report the average mapping error (ME), fMSE of LUT result, and the average time of calculating LUT by traversing all frames. The results are summarized in Table 3. Our method does not need to solve an optimization problem and works faster than "Op".

For both "Ours" and "Op", the mapping error becomes smaller when B increases because of the higher precision of LUT. The mapping error becomes larger when T increases, because one color value may be associated with multiple harmonized color values.

For direct optimization "Op", we can treat $\{\mathcal{C}, \tilde{\mathcal{C}}\}$ as the training set and the current frame as test set. The error on the training (*resp.*, test) set is ME (*resp.*, fMSE). When B increases from 32 to 128, ME decreases but fMSE increases, probably due to the overfitting issue. Similarly, larger T means larger training set and leads to better generalization ability. "Ours" shows the same tendency. One reason is that larger B and smaller T will produce more invalid pixels, resulting in worse fMSE.

5.6 Qualitative Comparison

Following [Haozhi *et al.*, 2019], we show two adjacent frames of the composite video, harmonized results by $\mathrm{iS^2AM}$ and Ours ($\mathrm{iS^2AM}$), and ground-truth video in Figure 3. It shows that our method can produce visually appealing harmonized results which are closer to the ground-truth. We also report the RGB values of two temporally identical pixels in two frames to show the temporal consistency. We can see the two pixels in $\mathrm{iS^2AM}$ have a large color discrepancy without considering temporal consistency, while Ours ($\mathrm{iS^2AM}$) can obtain temporally coherent results. More results are shown in Supplementary.

5.7 Evaluation on Real Composite Videos

The input composite videos used in previous sections are synthetic composite videos, which may have a domain gap with real composite videos. To synthesize real composite videos, we first collect 30 video foregrounds with masks from a video matting dataset [Sun *et al.*, 2021] as well as 30 video backgrounds from Vimeo-90k Dataset [Tianfan *et al.*, 2019] and Internet. Then, we create composite videos via copy-and-paste and select 100 composite videos which look reasonable *w.r.t.* foreground placement but inharmonious *w.r.t.* color/illumination. The harmonized results and user study results are left to Supplementary.

5.8 Hyper-parameter Analyses and Limitations

We investigate the impact of two hyper-parameters: the number of neighboring frames T and the number of bins B in the LUT. Additionally, we discuss the limitation of our method. The above results are left to Supplementary.

6 Conclusion

In this paper, we have contributed a new video harmonization dataset HYouTube which consists of pairs of synthetic composite videos and ground-truth real videos. We have also designed a novel framework $\mathrm{CO_2Net}$ based on color mapping consistency. Extensive experiments on our HYouTube dataset and real composite videos have demonstrated the effectiveness of our proposed framework.

Acknowledgements

The work is supported by the National Key R&D Program of China (2018AAA0100704), the Shanghai Municipal Science and Technology Major Project, China (2021SHZDZX0102), National Natural Science Foundation of China (Grant No.61902247).

References

[Bo *et al.*, 2011] Fan Bo, Zhou Fugen, and Han Hongbin. Medical image enhancement based on modified lut-mapping derivative and multi-scale layer contrast modification. In *International Congress on Image and Signal Processing*, 2011.

[Cong *et al.*, 2020] Wenyan Cong, Liu Liu Zhixin Ling Weiyuan Li Jianfu Zhang, Li Niu, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020.

[Cong *et al.*, 2021] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Bargainnet: Background-guided domain translation for image harmonization. In *ICME*, 2021.

[Cong *et al.*, 2022] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. *CVPR*, 2022.

[Cun and Pun, 2020] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020.

[Fischl and Schwartz, 1999] Bruce Fischl and Eric L Schwartz. Adaptive nonlocal filtering: a fast alternative to anisotropic diffusion for image enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):42–48, 1999.

[Guo *et al.*, 2021] Zonghui Guo, Zhaorui Gu Haiyong Zheng, Yufeng Jiang, and Bing Zheng. Intrinsic image harmonization. In *CVPR*, 2021.

[Hao *et al.*, 2020] Guoqing Hao, Satoshi Iizuka, and Kazuhiro Fukui. Image harmonization with attention-based deep feature modulation. In *BMVC*, 2020.

[Haozhi *et al.*, 2019] Huang Haozhi, Wei Liu Senzhe Xu, Junxiong Cai, and Shimin Hu. Temporally coherent video harmonization using adversarial networks. *IEEE Transactions on Image Processing*, 29:214–224, 2019.

[Kalyan *et al.*, 2010] Sunkavalli Kalyan, Wojciech Matusik Micah K. Johnson, and Hanspeter Pfister. Multi-scale image harmonization. *Acm Trans Graph*, 29(4):1–10, 2010.

[Konstantin *et al.*, 2021] Sofiiuk Konstantin, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. *WACV*, 2021.

[Lalonde and Efros, 2007] Jean-Francois Lalonde and Alexei A Efros. Using color compatibility for assessing image realism. In *ICCV*, 2007.

[Ling *et al.*, 2021] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *CVPR*, 2021.

[Liu *et al.*, 2020] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020.

[Mese Murat, 2001] Palghat P. Vaidyanathan. Mese Murat. Look-up table (lut) method for inverse halftoning. *IEEE Transactions on Image Processing*, 10(10):1566–1578, 2001.

[Michael *et al.*, 2006] Elad Michael, Boaz Matalon, and Michael Zibulevsky. Image denoising with shrinkage and redundant representations. In *CVPR*, 2006.

[Phillip *et al.*, 2016] Isola Phillip, Tinghui Zhou Jun-Yan Zhu, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2016.

[Reinhard *et al.*, 2001] Erik Reinhard, Bruce Gooch Michael Ashikhmin, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

[Shu *et al.*, 2015] Fujita Shu, Makoto Kimura Norishige Fukushima, and Yutaka Ishibashi. Randomized redundant dct: efficient denoising by using random subsampling of dct patches. In *Siggraph Asia Technical Briefs*, 2015.

[Su *et al.*, 2012] Xue Su, Julie Dorsey Aseem Agarwala, and Holly Rushmeier. Rushmeier h.: Understanding and improving the realism of image composites. In *Acm Trans Graph*, volume 31, pages 1–10, 2012.

[Sun *et al.*, 2021] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. Deep video matting via spatio-temporal alignment and aggregation. In *CVPR*, 2021.

[Tianfan *et al.*, 2019] Xue Tianfan, Donglai Wei Baian Chen, Jiajun Wu, and William T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[Xu *et al.*, 2018] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[Yi-Hsuan *et al.*, 2017] Tsai Yi-Hsuan, Kalyan Sunkavalli Xin Lu Xiaohui Shen, Zhe Lin, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017.

[Zhu *et al.*, 2015] Jun-Yan Zhu, Eli Shechtman Philipp Krahenbuhl, and Alexei A. Efros. Learning a discriminative model for the perception of realism in composite images. In *ICCV*, 2015.