HyperCUT: Video Sequence from a Single Blurry Image using Unsupervised Ordering — Supplementary Material —

Bang-Dang Pham[1,*] Phong Tran[1,2,*] Anh Tran[1] Cuong Pham[1,3] Rang Nguyen[1] Minh Hoai[1,4]

[1]VinAI Research, Vietnam [2]MBZUAI, UAE [3]Posts & Telecommunications Inst. of Tech., Vietnam [4]Stony Brook University, USA

{v.dangpb1, v.anhtt152, v.hoainm}@vinai.io cuongpv@ptit.edu.vn the.tran@mbzuai.ac.ae

[*]Equal contribution

Abstract

In this supplementary PDF, we first specify some techniques applied to our data post-processing. Then, we provide the details of our proposed HyperCUT architecture. Finally, we illustrate and analyze the qualitative results compared with the SOTA model.

1. Data post-processing

Random background for Face/Hand subsets. For the datasets collected using green screen, we use [1] to extract the foreground objects and then blend them into random backgrounds collected on the internet. Some examples are shown in Fig. 1.

Color correction algorithm. Given two reference images (x, y), we calculate the color correction matrix of the pair (The map $C_{x,y}(\cdot)$ in the paper) by minimizing:

$$M^* = \underset{M \in R^{3x4}}{\arg\min} \|Mx - y\| \tag{1}$$

M^* can be easily calculated by using linear regression.

Equation 11 in the paper explains how we can find reference images to calibrate color between two cameras. To find the frame index p in that equation, we simply iterate all possible position, and then choose the one i with largest $PSNR(C_{y_i^{fake}}, y)$. Details are given in Algorithm 1.

In addition, for the datasets collected in the laboratory (face and hand categories of our proposed dataset), before applying the color correction algorithm mentioned in the paper, we first calibrate the colors of the two cameras using a color checker.

Others. Due to the design of the camera system, after capturing images, we crop the border of the images to remove the black region of the cameras, letting the size of each image be 448×448.

Algorithm 1 Color correction

Input: y, $x[0], x[1], ..., x[h]$
Output: 7 calibrated sharp frames $x_0, x_1, ..., x_6$

$i \leftarrow 0$
$p \leftarrow -1$
$best \leftarrow -\infty$
while $i + 6 < h$ **do**
  $y_{fake} \leftarrow synBlur(x[i], x[i+1], ..., x[i+6])$
  $C \leftarrow ColorCorrectionMap(y_{fake}, y)$
  if $PSNR(C(y_{fake}), y) > best$ **then**
    $p \leftarrow i$
  end if
  $i \leftarrow i + 1$
end while
$x_0, x_1, ..., x_6 = x[p], x[p+1], ..., x[p+6]$

1.1. Additional dataset statistics

In the paper, we use a synthetic blur2vid dataset generated from REDS [2] for training and testing. Here we provide the numbers of training and testing sequences of the set in Table 1.

Table 1. Statistics of the synthetic datasets used in the paper

Dataset	#data samples	
	Train	Test
REDS	58876	1330

2. Details of the network architecture

The detailed architecture of our proposed network for $\mathcal{H}$ is shown in Table 2 with $n = 128$.

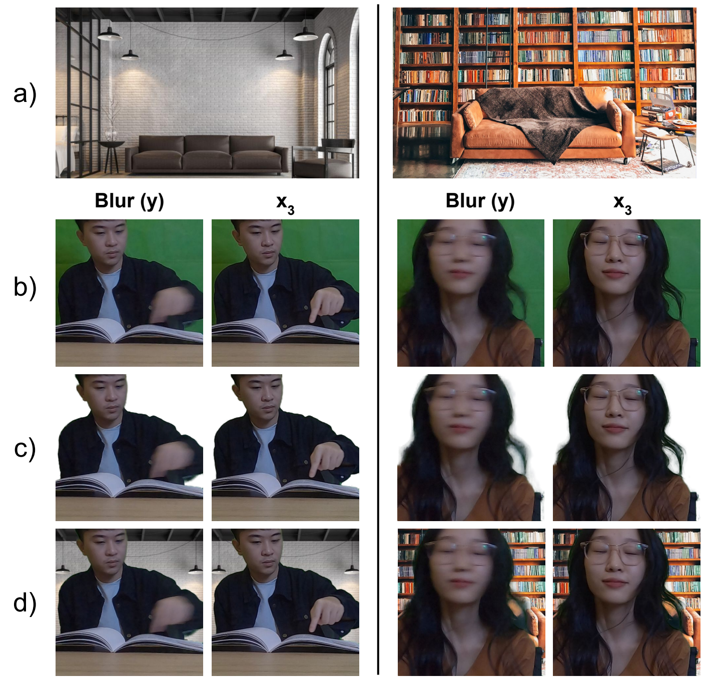

Figure 1. **Matting examples**. Row (a) are random background images collected on the Internet. Row(b) are images captured using green screen. Row (c) are extracted foreground using [3]. Row (d) are the final images used to train the models.

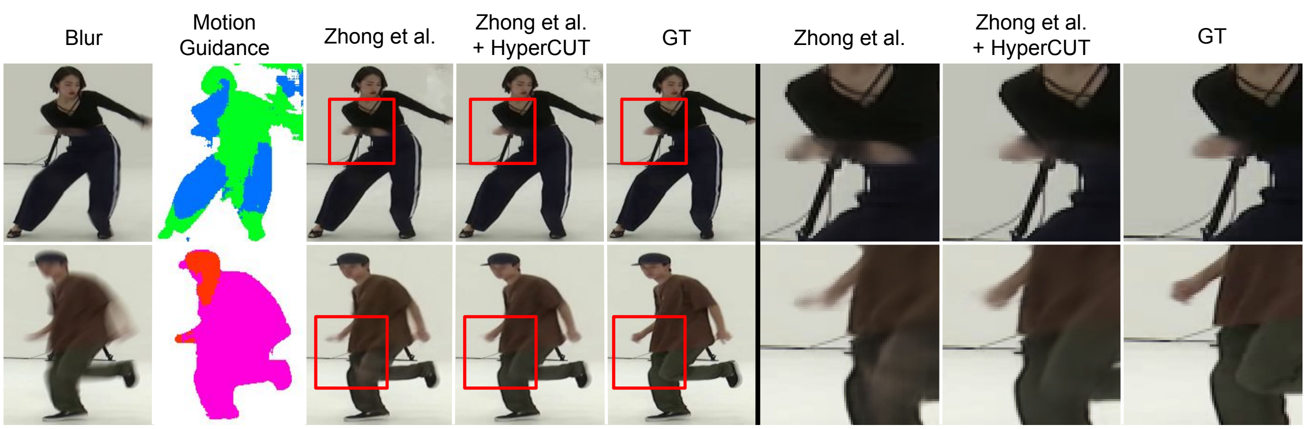

Figure 2. **Qualitative comparison with [4] in solving order-ambiguity.** We test Zhong et al. [4] models with the original setting and after embedding our regularization. The red region emphasizes the contribution of our HyperCUT module in overcoming the order-ambiguity issue of the model. For a fair comparison, we use the same motion guidance but predict with two different decomposer modules, one with the original loss [4] and one with HyperCUT regularization.

3. Additional Qualitative Results

3.1. Order-Ambiguity Impact

We test our embedding module on the Synthetic dataset B-Aist++ as mentioned in [4]. As shown in Fig. 2, the ordering proposed by our HyperCUT module helps the baseline model overcome the reconstruction issue to some extent when using the same motion guidance. Especially when blur is caused by fast movement, as given in the first row of Fig 2, the specific direction in the training state through HyperCUT regularization can improve the model stability.

3.2. Video Result

We present the video results as result.mp4. We first show an blurry image, then same sample from motion guidance prediction network, the original result of [4], result after embedding our HyperCUT module and the corresponding ground truth.

Layer	Output shape
	$H \times W \times 6$
ReflectionPad2d(3) Conv2d(6, 32, 7, 1, 0)	$H \times W \times 64$
Conv2d(32, 32, 3, 2, 1) LeakyReLU()	${}^{H}/_{2} \times {}^{W}/_{2} \times 32$
Conv2d(32, 64, 3, 2, 1) LeakyReLU()	${}^{H}/_{4} \times {}^{W}/_{4} \times 64$
Conv2d(64, 128, 3, 2, 1) LeakyReLU()	${}^{H}/_{8} \times {}^{W}/_{8} \times 128$
Conv2d(128, 128, 3, 2, 1) LeakyReLU()	${}^{H}/_{16} \times {}^{W}/_{16} \times 128$
Conv2d(128, 128, 3, 2, 1) LeakyReLU()	${}^{H}/_{32} \times {}^{W}/_{32} \times 128$
ResBlock(128, 128) $\times 6$	${}^{H}/_{32} \times {}^{W}/_{32} \times 128$
AdapativeAvgPool2d(1, 1)	128

Table 2. Network architecture

References

[1] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[2] Seungjun Nah, Radu Timofte, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

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

[4] Zhihang Zhong, Xiao Sun, Zhirong Wu, Yinqiang Zheng, Stephen Lin, and Imari Sato. Animation from blur: Multi-modal blur decomposition with motion guidance. In *Proceedings of the European Conference on Computer Vision*. Springer, 2022.