Efficient Video Matting on Human Video Clips for Real-Time Application

Chao-Liang Yu
College of Computer Science
National Yang Ming Chiao Tung University
Hsinchu City, Taiwan
daveyu824.cs09@nycu.edu.tw

I-Chen Lin
College of Computer Science
National Yang Ming Chiao Tung University
Hsinchu City, Taiwan
ichenlin@cs.nycu.edu.tw

***Abstract*—This paper presents an efficient and effective matting framework for human video clips. To alleviate the inefficiency problem in existing models, we propose using a refiner dedicated to error-prone regions, and reduce the computation at higher resolutions, so the proposed framework can achieve real-time performance for 1080p 60fps videos. Also, with the recurrent architecture, our model is aware of temporal information and produces temporally more consistent matting results compared to models processing each frame individually. Moreover, it contains a module for capturing semantic information. That makes our model easy to use without troublesome setup, such as annotating trimaps or other additional inputs. Experiments show that our proposed method outperforms previous matting methods, and reaches the state of the art on the VideoMatte240K dataset.**

***Index Terms*—Video matting, refinement network, recurrent network, real-time processing**

I. Introduction

Given an input image I, the goal of matting is to extract the foreground F_i (i denotes the pixel index), and the alpha matte α_i. Then, we can compose a new image $I^{'}$ with a new background $B^{'}$ as follows:

$$I_i^{'} = \alpha_i F_i + (1 - \alpha_i) B_i^{'} \tag{1}$$

Traditionally, to estimate alpha mattes with satisfactory quality, an image has to be taken in front of a green screen, and this method is widely used in movie and news industry. However, it cannot be applied to general images. There have been both optimization-based algorithms and learning-based algorithms tackling this task, but most of them require either pre-defined trimaps or user-drawn scribbles as additional constraints to guide the matting evaluation. It is troublesome and costly to acquire and annotate these additional cues in advance, thus making it difficult to apply these methods in real-time applications.

To relieve such a cumbersome process, we first employed a simple encoder-decoder-based network to predict coarse semantic masks of the input video. These semantic masks bring similar benefits to guide the matting network about the semantic information of foreground objects, but they do not need to be annotated in advance. The mask extraction causes only minor increase in computing time.

Besides, our target is video matting. We cannot directly apply matting methods designed for a single image, because any inconsistency in alpha mattes across frames can be conspicuous. Fusing temporal information into a matting model brings several benefits. For example, because the model now considers multiple frames simultaneously, the predicted results should be more coherent. Also, when colors between the foreground objects and the background in a certain frame are too similar to distinguish, the model can refer to the features in adjacent frames to make the prediction more precise. Hence, we followed [1] to use a recurrent architecture to make our model aware of temporal information. The recurrent architecture not only makes output alpha mattes coherent across frames, but also improves the quality of each individual frame.

Another important issue is efficiency. Applying deep convolutional network models on high-resolution images usually takes long processing time and requires a lot of memory. The high power consumption also prevents users from applying these models in real-time application. Inspired by PointRend segmentation [2], we propose a PointRend-based refiner to focus on error-prone regions and save unnecessary computation. Our model first performs on a downsampled input to generate the low-resolution output, and the low-resolution alpha matte is then upsampled to the originally resolution with light-weight computation. The alpha values around error-prone regions like edges of human bodies are further corrected by the proposed refiner. Our method can save significant amount of computation, while maintaining comparable qualities with related methods.

In summary, we presented a matting model specifically for human videos, which incorporates a recurrent mechanism and does not need any external input. Its awareness of uncertain regions makes it efficient and accurate. Our method outperforms related methods in all representative metrics on VideoMatte240K [3] dataset, and produces new state-of-the-art results. It runs over 80 frames per second (FPS) on 1080p videos with an NVIDIA RTX 3090 GPU, which is considered real-time for most applications, such as video conferencing.

This work was partially supported by the Ministry of Science and Technology, Taiwan under grant no. MOST 109-2221-E-009-122-MY3.

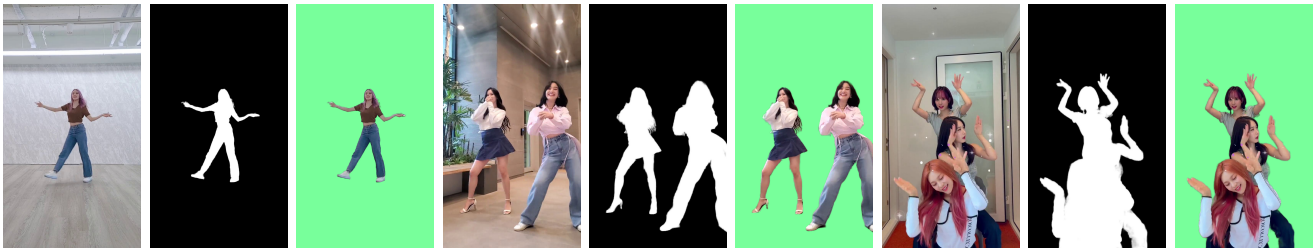


Fig. 1: Our model can extract alpha values and change backgrounds for video clips in real time without additional trimaps. The estimated alpha mattes are of less artifacts. (From the left to the right: inputs, alpha mattes, background-substituted images)

II. Related Work

As matting is a ill-posed problem, related methods usually rely on additional information like trimaps [4]–[6], or pre-taken background images [3], [7] to help the model better understand the structure of input images. Though these additional information are indeed helpful in producing better matting results, recently, researchers have paid attention on auxiliary-free methods [8]–[10] to make the whole matting process easier to use.

Several methods [11]–[13] have been proposed to make matting for video clips possible. To ensure temporal coherence across frames, they usually took and propagated information from one frame to another frame, instead of treating each frame individually. When deep learning has not become popular, temporal coherence is often achieved by utilizing optical flow, but the performance is limited by the quality and the computing time of optical flow evaluation. Lin et al. [1] used a recurrent decoder with ConvGRU to capture dependencies across frames, whereas Wang et al. [14] utilized a graph neural network along with deformable convolution layers for the same purpose.

Recurrent architectures like LSTM (Long Short-Term Memory) [15] and GRU (Gated Recurrent Unit) [16] have been widely used to deal with sequential data, and their variants ConvLSTM [17] and ConvGRU [18] have also been proposed. Because videos are sequences of images, and naturally fit the goal of recurrent architectures, we followed [1] to adopt ConvGRU for our task.

Two impressive works [19] and [2] have been proposed to speed up deep learning models on high resolution images. Both of their core ideas is to run the model on lower resolution first, and upsample the intermediate output with auxiliary hidden features to correct errors caused by direct upsampling. Similar techniques have also been used on matting tasks in [3] and [1]. We adapted a PointRend-based refiner for our work because it is more effective and can be configured for various computing capability.

III. Method

We first summarize several problems in current matting methods, and our corresponding designs. First, annotating trimaps is very laborious for video clips, but most matting methods rely on these auxiliary input to produce better results. Our framework roughly predicts the contour of foreground objects, so the overall model can still benefit from additional semantic information, but it does not need to be provided by users in advance. Secondly, videos are sequential data, and there are coherence between frames. By introducing recurrent architectures, our network can learn from past information and ignore irrelevant noises that appear in only few frames. Lastly, efficiency is critical in real-time application, and the computation becomes more intensive at high resolutions. To save a significant amount of computation, our method performs the primary prediction at low resolutions, and it focuses on and refines uncertain regions at the high resolution.

Fig. 2 shows our model overview. Our model includes a mask-prediction network for extracting semantic information, a matting network for generating low-resolution alpha mattes and foregrounds, and a PointRend-based refiner for correcting error-prone regions of the output after direct upsampling.

A. PointRend-based Refiner

We investigated the cause of low efficiency of existing matting methods, and found that their models exhaustively evaluate regions throughout the whole image. However, it is indeed unnecessary for human video matting, since ambiguous translucency usually occurs at a few locations, e.g. around the human contour and hair. Therefore, the key idea of our model design is to first evaluate the alpha matte in a low resolution. In the following upsampling, the presented refiner focuses only on error-prone regions to save computation.

Fig. 3 shows the overall concept of the proposed PointRend-based refiner. At each stage, the image is first bilinearly upsampled, and the top k points that need refinement most are selected. These points are concatenated with hidden features before sending into the refiner module to produce fine-grained outputs. To make the refining process efficient, the module simply consists of a Conv1D layer, and shares weight across multiple upsampling stages.

As the proposed refinement network only works on error-prone regions, it is important to define what is the so-called "error-prone regions". To this end, we propose three different criterion strategies for selecting error-prone regions.

***Incoherence* as the criterion:** This strategy assumes that outputs alpha values should be coherent both spatially and temporally, so regions with severe changes compared to others should be considered candidates for refinement. We applied Laplacian filter on input alpha frames to get spatial incoherence. For temporal incoherence, at timestamp t, we

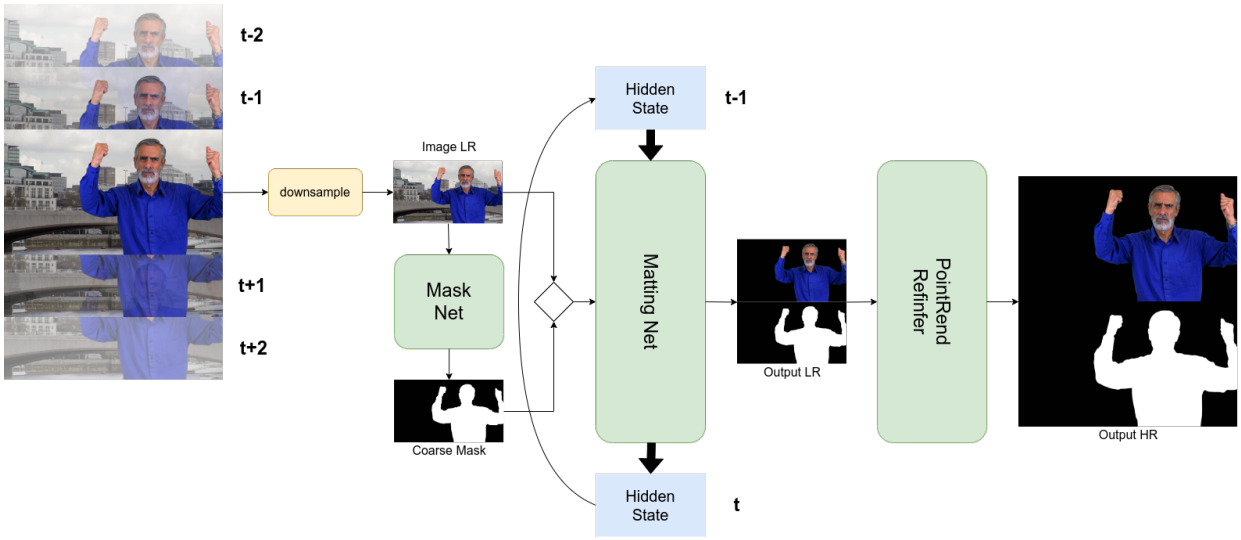


Fig. 2: Flowchart of the proposed system. An input frame is first downsampled by a factor of s and fed into the mask-prediction network, and the output semantic mask is concatenated with the input frame before sent into the main matting network. Finally, the alpha matte at the original resolution is estimated through the PointRend-based refiner. Outputs from the ConvGRU modules of the matting network at the current timestamp t act as the hidden state for the next timestamp.

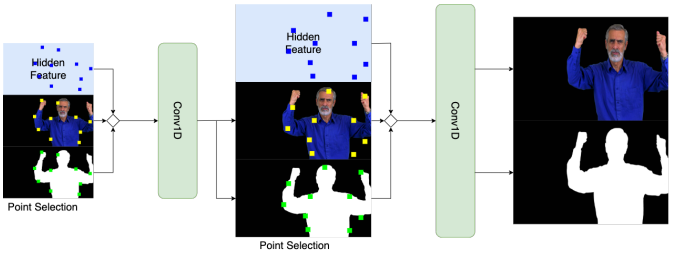


Fig. 3: The conceptual diagram of our refinement network. It refines only the error-prone regions, and the rest regions are directly upsampled. A few points are marked as the error-prone locations mentioned above.

calculated $|\alpha_i^t - \alpha_i^{t-1}|$, and summed them up to get the final uncertainty score.

***Predicted error map* as the criterion:** Originally, the last layer of our matting network generates 4-channel outputs, which include an alpha value and three foreground color values. We made the model output an extra channel as an error map. We defined the ground truth of e_i, the error at pixel i, to be the difference between alpha prediction α_i and alpha ground truth α_i^*. It was trained using the following loss function:

$$L^{error} = \|e - (\alpha - \alpha^*)\|_2 \tag{2}$$

***Distance to the alpha valley point* as the criterion:** We noticed that different from objects such as glasses, plastic bags, and water, most regions of alpha mattes for human images are opaque, i.e., most of their alpha values are either 0 or 1 (or 255). To validate our assumption, we extracted the boundary regions of alpha mattes in the VideoMatte240K [3] dataset, and calculated the frequency of each possible pixel value. The distribution of alpha values ([0, 255]) of VideoMatte240K is shown in Fig. 4. We argued that pixels near the valley point of the alpha distribution should be assigned larger uncertainty and is worth careful inspection. We assigned the uncertainty score u_i for a pixel with alpha value α_i as $u_i = |\alpha_i - \alpha_{vp}|$. ($\alpha_{vp}$ is 0.5 in our case when alpha values are scaled to 0∼1.)

As discussed later in the experiment and the supplementary material, our propose framework adopts the *distance to alpha valley point* as the criterion for error-prone regions.

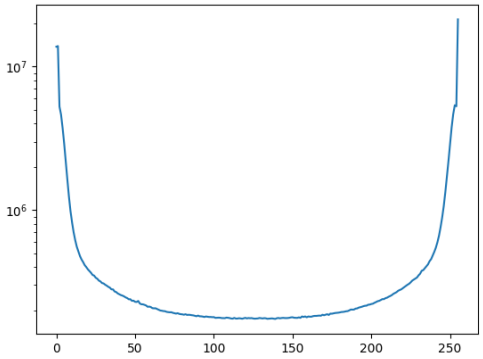


Fig. 4: Distribution of alpha values in VideoMatte240K. The horizontal axis represents the alpha values [0,255], and the vertical axis represents the number of pixels in log scales.

B. Matting Network

The overall architecture of our matting network is shown in Fig. 5. We adopted MobileNet V3 [20] pretained on ImageNet

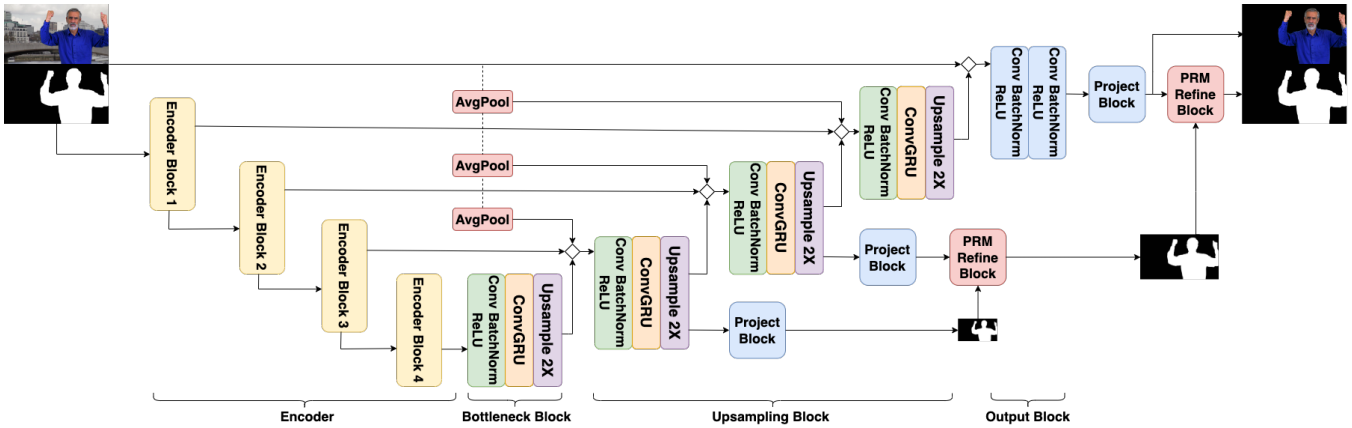


Fig. 5: The matting network follows the regular encoder-decoder architecture, where the decoder captures temporal information with ConvGRU, and we also employed the PRM modules to estimate more precise alpha outputs.

[21] as our backbone because it reaches a balance between speed and accuracy. It is followed by a LR-ASPP module [20], which is capable of mixing features from multiple resolutions. Feature maps at $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ scales are generated and sent into decoders for further processing.

Following [1], we also adopted ConvGRU in our decoder stage, so our model is able to learn and keep temporal information. Although compared to temporal attention mechanism, recurrent architecture lacks certain parallelism, but it can dynamically learn and forget past information, which suits video tasks. It also consumes less memory than temporal attention maps.

As shown in Fig. 5, the decoder stage can be divided into three categories: a bottleneck block, three upsampling blocks, and an output block. We applied PRM (Progressive Refinement Module) [22] to further enhance the accuracy of alpha values. First, decoder blocks at $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{1}$ scales are followed by a projection block to generate alpha mattes with different sizes. They are progressively refined in a coarse-to-fine manner, where confident regions in lower levels are preserved, and non-confident regions are replaced with outputs from higher levels. Please refer to the supplementary material for details about ConvGRU and PRM modules.

C. Mask-Prediction Network

The structure of mask-prediction network (MaskNet) is similar to the matting network structure, but ConvGRU layers and RPM modules are removed to make the network lighter. The projection block at the end of the output block is also replaced with a simple Conv2D layer for channel reduction. Because we only need the contour of the foreground object, and accuracy is not the top priority here, to further speed up computation, input frames are once again downsampled by a factor of 2. Because the downsampling operation may lead to information loss, which is unfavorable for high-level features, to prevent the rough semantic masks from interfering with convolution operations, we do not send semantic masks into the output block via skip connection in the matting network. Due to page limitation, please refer to the supplementary material for the loss functions applied in our framework.

IV. Experiment

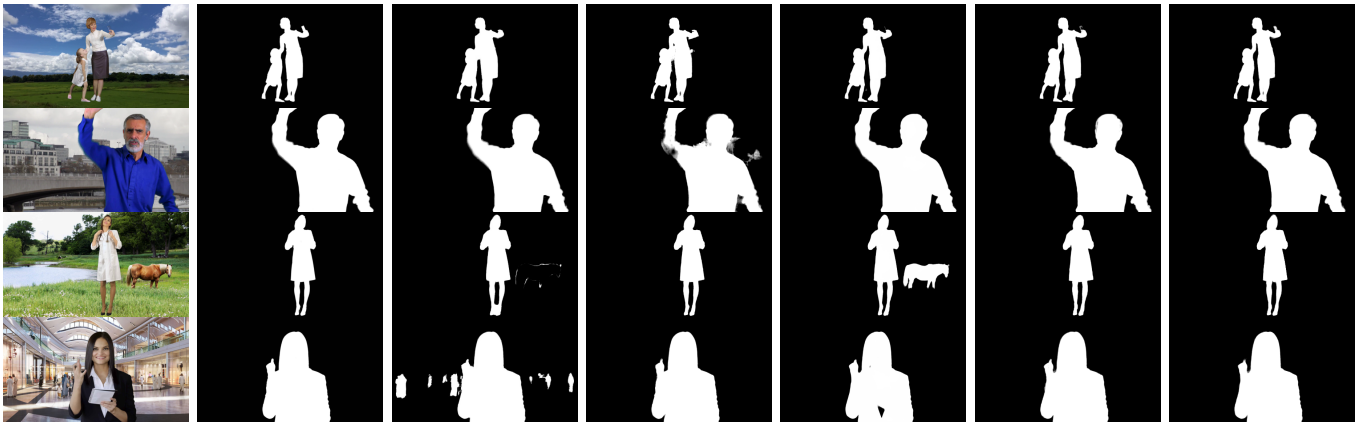
We evaluated our model on three datasets including VideoMatte240K (VM240K) [3], Adobe Image Matting (AIM) [4], and Distinctions-646 (D646) [8]. This section first compares our results with those by other matting methods and then reports the ablation studies. Several failure cases of our framework are also shown in the end of section. The detail of the datasets we used, how we conducted the training process, and more results are shown in the supplementary material.

A. Evaluation

We compared our model against related state-of-the-art methods, including MatteFormer [23], BGMv2 [3], MODNet [9], and RVM [1] (auxiliary-free). MatteFormer requires finely annotated trimaps, and BGMv2 needs the background image to be well aligned to produce the best result, whereas we focus on auxiliary-free matting. For fair comparison, MatteFormer is provided with pseudo trimaps generated by DeepLabV3 [24] with ResNet-101 [25] backbone, and BGMv2 only sees the background of the first input frame. We performed all the methods with the official weights shared by their authors.

We followed the same approach as [1] to conduct the evaluation, where each test sample from VM240K, AIM, and D646 was composited onto 5 video and 5 image backgrounds. Image datasets were applied with motion augmentations. For VM240K, we used the test data shared by [1]. For AIM and D646, we had to composite test data by ourselves because these two datasets are not publicly available. However, as backgrounds are randomly sampled, we cannot 100% reproduce the result as in [1]. In Table I, we mark related methods performed on our data with *.

For alpha outputs, we evaluated them with MAD(mean absolute difference), MSE(mean squared error), Grad(spatial gradient) [26], Conn(connectivity) [26], and dtSSD [27]. The former four metrics measure the quality of each individual frame, whereas dtSSD measures temporal coherence.



(a) Input (b) GT (c) MatteFormer (d) BGMv2 (e) MODNet (f) RVM (g) Ours

Fig. 6: Qualitative comparison on VM240K SD dataset. Compared with others, our method can precisely locate the foreground object, and is stable and consistent on various kinds of input data.

Dataset	Method	Alpha MAD	Alpha MSE	Alpha Grad	Alpha Conn	Alpha dtSSD	FG MSE
VM240K 512×288	MatteFormer	8.13	3.67	2.24	0.70	2.25	
	BGMv2	25.19	19.63	2.28	3.26	2.74	
	MODNet	9.41	4.30	1.89	0.81	2.23	
	RVM	6.08	1.47	0.88	0.41	1.36	
	Ours	**5.50**	**0.91**	**0.74**	**0.32**	**1.35**	
AIM 512×512	BGMv2	44.61	39.08	5.54	11.60	2.69	3.31
	MODNet	21.66	14.27	5.37	5.23	1.76	9.51
	RVM	14.84	8.93	4.35	3.83	1.01	5.01
	MatteFormer*	42.98	36.15	12.08	11.15	3.13	14.41
	RVM*	19.82	12.28	5.93	5.18	1.31	6.50
	Ours	18.09	10.51	6.45	4.73	1.58	5.57
D646 512×512	BGMv2	43.62	38.84	5.41	11.32	3.08	2.60
	MODNet	10.62	5.71	3.35	2.45	1.57	6.31
	RVM	7.28	3.01	2.81	1.83	1.01	2.93
	MatteFormer*	21.96	17.56	9.23	5.65	2.67	4.73
	RVM*	7.89	3.45	2.86	1.97	1.00	2.92
	Ours	6.90	2.65	2.78	1.70	1.03	3.57

TABLE I: Quantitative comparison on low-resolution outputs. Our method outperforms related methods on VM240K in all metrics, and performs among the top on AIM and D646.

Dataset	Method	MAD	MSE	Grad	dtSSD
VM240K 1920×1080	RVM	6.57	1.93	10.55	1.9
	Ours	**6.49**	**1.88**	**10.48**	**1.86**
AIM 2048×2048	RVM	19.31	11.99	**47.65**	**1.49**
	Ours	**18.17**	**10.84**	51.95	1.74
D646 2048×2048	RVM	8.98	4.54	**31.22**	**1.77**
	Ours	**8.06**	**3.73**	32.54	1.87

TABLE II: Quantitative comparison on high-resolution outputs. Our method outperforms RVM in most metrics.

Table I and Fig. 6 compare our method against others on low-resolution input. Note that the refiner is not used here. It can be clearly seen that both MatteFormer and BGMv2 must rely on additional information (trimaps & backgrounds) to produce accurate output, and do not perform satisfactorily with pseudo trimaps and dynamic backgrounds. Our method outperforms MODNet and RVM with less semantic-level error and more accurate alpha mattes.

Table II further compares our method against RVM on high-resolution input. In such cases, RVM uses [28] for upsampling and filtering. For our refiner, we used downsampling factor $s = 0.25$, and selected $k = (h/8) * (w/8)$ points for refinement. Our method is not only better than RVM in most metrics, but also avoid weird artifacts at boundary regions. Please refer to the supplementary video for visual comparison.

Method	MAD	MSE	Grad	dtSSD
w/o coarse mask	6.56	1.88	**10.44**	1.88
w/o temporal information	6.74	1.93	10.95	2.21
Ours	**6.49**	1.88	10.48	**1.86**

TABLE III: Studies on effectiveness of network components.

Method	MAD	MSE	Grad	dtSSD
Incoherence	6.89	2.07	11.59	1.98
Error map	7.11	2.03	12.54	1.99
Ours (valley point)	**6.49**	**1.88**	**10.48**	**1.86**

TABLE IV: Studies on strategies to select error-prone regions.

B. Ablation Studies

We conducted ablation studies to discuss the effectiveness of components in our network. We first removed the mask-prediction network, and passed zero tensors into the ConvGRU modules to take away temporal information. The statistics are shown in Table III.

We also compared different strategies for selecting regions for refinement. From Table IV, it is clear that our method, which regards error-prone regions as points with alpha values close to the valley point (0.5 in [0, 1] range), significantly outperforms the other two strategies.

C. Failure Cases

Several failure cases are shown in Fig. 7. Our model may fail when there are too many people overlapped in the scene. It also can not generate precise results if the scene is highly complicated, e.g. mixture of human motion blur and translucent bubbles.

Fig. 7: Failure cases.

V. Conclusion

This paper presents a novel matting model for human video clips. Our method does not require any auxiliary input, and considers temporal information to produce more coherent results. The proposed refiner architecture significantly reduces computation and makes our method able to run in real-time on 1080p 60fps videos with high accuracy. Experiments show that our method reaches state of the art on video matting. In the future, we are interested in further enhancing the concept of error-prone selection into the framework and loss functions [29].

References

[1] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta, "Robust high-resolution video matting with temporal guidance," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 238–247.

[2] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9799–9808.

[3] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8762–8771.

[4] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang, "Deep image matting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2970–2979.

[5] Marco Forte and François Pitié, "f, b, alpha matting," *arXiv preprint arXiv:2003.07711*, 2020.

[6] Rui Wang, Jun Xie, Jiacheng Han, and Dezhen Qi, "Improving deep image matting via local smoothness assumption," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[7] Junjie Deng, Yangyang Xu, Zeyang Zhou, and Shengfeng He, "Background matting via recursive excitation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[8] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei, "Attention-guided hierarchical structure aggregation for image matting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13676–13685.

[9] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau, "Modnet: Real-time trimap-free portrait matting via objective decomposition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 1140–1147.

[10] Linhui Dai, Xiang Song, Xiaohong Liu, Chengqi Li, Zhihao Shi, Martin Brooks, and Jun Chen, "Enabling trimap-free image matting with a frequency-guided saliency-aware network via joint learning," *IEEE Transactions on Multimedia*, 2022.

[11] Nicole Brosch, Asmaa Hosni, Christoph Rhemann, and Margrit Gelautz, "Spatio-temporally coherent interactive video object segmentation via efficient filtering," in *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*. Springer, 2012, pp. 418–427.

[12] Dingzeyu Li, Qifeng Chen, and Chi-Keung Tang, "Motion-aware knn laplacian for video matting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3599–3606.

[13] Ehsan Shahrian, Brian Price, Scott Cohen, and Deepu Rajan, "Temporally coherent and spatially accurate video matting," in *Computer Graphics Forum*. Wiley Online Library, 2014, vol. 33, pp. 381–390.

[14] Tiantian Wang, Sifei Liu, Yapeng Tian, Kai Li, and Ming-Hsuan Yang, "Video matting via consistency-regularized graph neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4902–4911.

[15] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[17] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[18] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[19] Baoguang Shi, Xiang Bai, and Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[20] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al., "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[22] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille, "Mask guided matting via progressive refinement network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1154–1163.

[23] GyuTae Park, SungJoon Son, JaeYoung Yoo, SeHo Kim, and Nojun Kwak, "Matteformer: Transformer-based image matting via prior-tokens," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11696–11706.

[24] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott, "A perceptually motivated online benchmark for image matting," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 1826–1833.

[27] Mikhail Erofeev, Yury Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Jue Wang, "Perceptually motivated benchmark for video matting," in *BMVC*, 2015, pp. 99–1.

[28] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang, "Fast end-to-end trainable guided filter," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1838–1847.

[29] Wei-Lun Huang, Chun-Yi Hung, and I-Chen Lin, "Confidence-based 6d object pose estimation," *IEEE Transactions on Multimedia*, vol. 24, pp. 3025–3035, 2022.