Deep Video Matting via Spatio-Temporal Alignment and Aggregation

Yanan Sun
HKUST
now.syn@gmail.com

Guanzhi Wang
Stanford
guanzhi@cs.stanford.edu

Qiao Gu
CMU
qiaog@andrew.cmu.edu

Chi-Keung Tang
HKUST
cktang@cs.ust.hk

Yu-Wing Tai
Kuaishou Technology, HKUST
yuwing@gmail.com

Abstract

Despite the significant progress made by deep learning in natural image matting, there has been so far no representative work on deep learning for video matting due to the inherent technical challenges in reasoning temporal domain and lack of large-scale video matting datasets. In this paper, we propose a deep learning-based video matting framework which employs a novel and effective spatio-temporal feature aggregation module (ST-FAM). As optical flow estimation can be very unreliable within matting regions, ST-FAM is designed to effectively align and aggregate information across different spatial scales and temporal frames within the network decoder. To eliminate frame-by-frame trimap annotations, a lightweight interactive trimap propagation network is also introduced. The other contribution consists of a large-scale video matting dataset with groundtruth alpha mattes for quantitative evaluation and real-world high-resolution videos with trimaps for qualitative evaluation. Quantitative and qualitative experimental results show that our framework significantly outperforms conventional video matting and deep image matting methods applied to video in presence of multi-frame temporal information.

1. Introduction

Video matting, or extracting from a given video high-quality alpha mattes of a moving foreground object, has a wide range of applications in special effect and TV/movie production. Formally, given the color of a video frame I, foreground color F, background color B and alpha matte $\alpha \in [0, 1]$, the video compositing equation Eq. 1 is

$$I = \alpha F + (1 - \alpha)B. \tag{1}$$

Compared to image matting, video matting poses two further challenges. First, video matting needs to preserve spatial and temporal coherence in the predicted alpha mattes. A straightforward solution applying image matting on individual frames may inevitably cause severe flickering artifacts for moving fine details. Using optical flow to regularize output may help to alleviate these artifacts, but even with the most state-of-the-art optical flow estimation methods [1, 2, 3], optical flow estimation within complex matting regions is still very unreliable. This is because matting regions simultaneously contain both the foreground and background and there is so far no good optical flow estimation that can handle large area of semi-transparency.

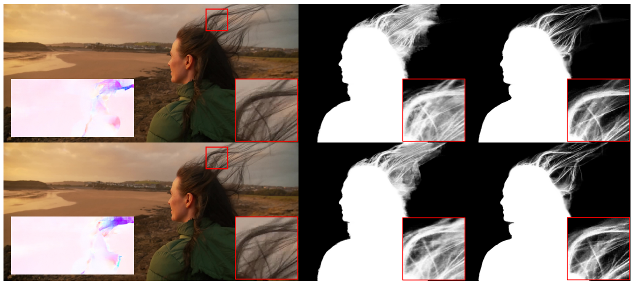


Figure 1. A challenging video matting example where top and bottom are consecutive frames. Left: input frames with insets zooming in complex hairs and showing the estimated optical flow from PWC-Net [2]. Note that the estimated optical flow is unreliable within the hair regions. Middle: deep image matting [6] results. Right: our results.

Traditional methods tackled the video matting problem by finding local or non-local affinity among pixel colors and computing the motion of the foreground [4, 5] but their results are still far from satisfactory especially when dealing with complex cases, such as rapidly moving objects or complex backgrounds. Figure 1 shows an example with challenging motions. The other challenge for video matting is the necessary input of a dense trimap for each frame, making it difficult to generate high quality large-scale video matting benchmarks.

In this paper, we propose an encoder-decoder network consisting of a novel spatio-temporal feature aggregation module (ST-FAM) for extracting feature pyramids at different levels, which utilizes spatial and temporal information across multiple frames. *Without* optical flow estimation, our network can effectively address the video matting problem and produce spatially and temporally coherent alpha mattes, and can generate good predictions of hard cases using the fused temporal information. To provide reliable frame-by-frame trimaps with minimum user inputs, a novel correlation layer is introduced to propagate trimaps across different frames. With our trimap propagation method, a user can edit and propagate trimaps at an interactive frame rate.

To support our and future video matting research, we have also contributed a high-quality video matting dataset with groundtruth alpha mattes. Furthermore, to verify the generalization capability of our method to real videos, we provide 10 high-resolution real-world videos with dense and frame-by-frame human annotated trimaps for evaluation. We evaluate our method on our composited test set as well as real-world high-resolution videos. Experimental results demonstrate that our deep video matting method significantly outperforms image-based deep matting methods and conventional video matting approaches, capable of handling complex scenarios such as rapidly moving objects with fuzzy boundaries or complex backgrounds.

2. Related Work

2.1. Image Matting

Traditional methods on natural image matting mainly use color and other relevant low-level image features for estimating alpha mattes via sampling, propagation or a combination of both. Sampling-based methods [7, 8, 9, 10, 11] first sample pixels from the foreground and background in a given image to construct pertinent color models which are used to estimate alpha values in the transition region. In propagation-based methods [12, 13, 14, 15, 16, 17, 18], Eq. 1 is reformulated so that alpha values are allowed to propagate from known foreground and background into the unknown transition region. Please refer to [19] for a comprehensive review on traditional matting methods.

For deep-learning based image matting, Cho *et al*. [20] proposed to apply deep neural networks to combine the complementary advantages of the results respectively produced by closed-form matting [17] and KNN matting [15]. Xu *et al*. [6] proposed a two-stage encoder-decoder network followed by a refinement network to address the image matting problem. Lutz *et al*. [21] proposed a generative adversarial network where dilated convolutions are integrated into the encoder-decoder network for improving matting performance. Wang *et al*. [22] introduced deep neural networks to learn an alpha matte propagation principle. Recently, a number of works have been proposed focusing on relaxing trimap input. Shen *et al*. [23] used an average shape mask for portraits to guide the network to infer the foreground and background regions automatically. Following the similar idea, Zhu *et al*. [24] designed a smaller portrait matting network and a fast filter that can be run in real time. Chen *et al*. [25] further eased the need for trimap on human, where a segmentation network is first used to predict foreground, background and transition regions from the input image. These regions are then fed together into another network to predict alpha mattes. Zhang *et al*. [26] extended this idea to general objects. They used a CNN with two decoders for foreground and background classification, and the classification results are then fed into a fusion network to obtain the alpha matte. Liu *et al*. [27] leveraged coarse annotated data with fine annotated data for boosting human matting without trimaps. They applied a mask prediction network taking hybrid data for generating human mask, a quality unification network for aligning the masks, and finally a matting refinement network for predicting the alpha mattes. Qiao *et al*. [28] further improved the performance of general object matting without trimap via attention mechanism. They proposed an end-to-end hierarchical attention network exploiting spatial and channel-wise attention to utilize appearance cues in a novel fashion. Sengupta *et al*. [29] introduced a different input setting. Their framework takes two photos with and without the foreground object as input to reduce trimaps labeling labor and provide external clues for model at a low cost.

2.2. Video Matting

Temporal coherency consideration is important for generating high-quality video mattes. Chuang *et al*. [30] used forward and backward optical flow to interpolate frame-wise trimaps and applied Bayesian matting to produce high-quality mattes of moving objects. Lee *et al*. [31] extended robust matting [32] by regarding time as a third spatial dimension and developed an anisotropic kernel using optical flow. Bai *et al*. [33] used a temporal matte filter to improve temporal coherence while preserving matte structures on individual frames. Choi *et al*. [4] used multi-frame non-local matting Laplacian in spatial temporal domain. Li *et al*. [34] implemented motion-aware KNN Laplacian to improve clustering of moving foreground pixels. Zou *et al*. [5] introduced a sparse and low-rank representation to construct non-local structure which yields better video matting results in terms of spatial and temporal consistency. A number of works tackled the video matting problem using custom hardware systems in video capture. Neel *et al*. [35] refocused images taken by a camera array to automatically generate alpha mattes for all video frames. McGuire *et al*. [36] computed alpha mattes from synchronized video streams taken by multiple cameras from the same point of view but with varying focus.

While there exist many traditional methods tackling

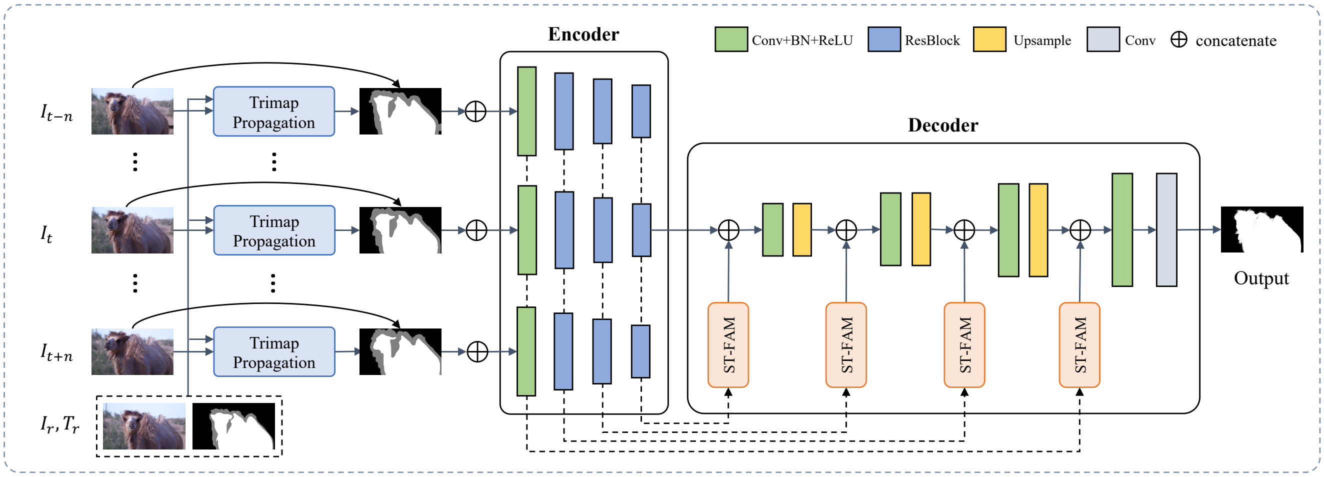


Figure 2. Our video matting framework. The lightweight trimap propagation network generates trimap of target frame I_t according to the trimap of reference frame I_r. The spatio-temporal feature aggregation module (ST-FAM) analyzes and fuses information at different levels from neighboring frames $\{I_{t-n}, \cdots, I_{t+n}\}$ and progressively outputs aggregated features to be fed into corresponding layers of the decoder. The structures of trimap propagation module and ST-FAM are shown in Figure 3.

video matting, their results are not as good as the recent deep learning-based methods, because the deep structural and semantic features are superior to low-level color features in traditional methods. In tackling spatial temporal coherency, later methods [4, 34, 5] that utilized non-local matting Laplacian to encode coherency have demonstrated better performance than the earlier post-processing based methods [30, 31, 33]. This is because the non-local matting Laplacian not only models the pertinent motions but also non-local similarities across different patches in different frames. We therefore believe that a deep network that can model the coherency inside its architecture to aggregate different scale of spatio-temporal features should outperform methods using straightforward post-processing.

3. Datasets

Our video matting datasets consist of well-selected composited and real-world videos which will be released together with the paper.

3.1. Composited Dataset

While there exist high-quality and large-scale datasets for image matting [37, 6], only a few video matting datasets with ground truth alpha mattes are available which are not suitable for training deep neural networks due to their limited sizes [38]. Thus, we create a new video matting dataset, which is composed of real foreground videos, their groundtruth alpha mattes, and background videos of a great variety of natural and real-life scenes.

Our foreground objects are made of both images and videos. For foreground video objects, we collect available green screen video clips from the Internet, from which we extract foreground colors and alpha mattes using a chroma keying software provided by Foundry Keylight [39]. Since the background of these videos is clean and simple, it is easy to estimate the accurate foregrounds and alpha mattes. Additionally, we also include high-quality images with groundtruth alpha mattes from the Adobe Deep Matting dataset [6] as foreground images. We discard similar images of the same object and keep 325 images out of 431 training samples and all 50 testing images.

The background set consists of various real-life videos. We collect over 6500 free video clips of natural scenarios, city views and indoor environment from the Internet. Most of these background videos are HD videos, and a few of them are 4K videos. We composite the foreground video and images onto the background videos using Eq. 1. During the composition, random and continuous translation, rotation and zooming are applied onto the foreground objects to simulate real videos containing a moving foreground.

Specifically, we composite each foreground object from 325 images and 75 videos with 16 randomly selected background videos, which generates 6400 videos as the train set. For the test set, we similarly combine each object from 50 images and 12 videos with 4 background videos, thus generating 248 test samples. The train and test sets are disjoint. To reduce memory and time cost for training and testing, each composited video contains at most 150 frames with the long side no more than 1920p. Compared to other video datasets, our dataset covers more varieties of video matting and provides rapidly-moving objects, which poses greater challenges for video matting evaluation.

3.2. Real-World High-Resolution Videos

In addition to the composited dataset, we provide 10 real-world videos at 4K resolution to evaluate the generalization ability of our video matting method. These videos are carefully selected from the Internet, consisting of various objects with large motions and complex scenes of real-life including humans, animals, plants, etc. For each video, we provide both fine and coarse dense trimaps for each frame.

4. Method

Figure 2 shows the overall framework, which consists of a lightweight trimap propagation module and a multi-frame encoder-decoder network. The trimap propagation module predicts the trimap for a target frame given a reference frame with trimap. The encoder extracts features at multiple levels from the input frames and trimaps. The decoder consists of a number of spatio-temporal feature aggregation modules (ST-FAM) which integrate deep features in neighboring frames to enhance alpha prediction of target frame. Figure 3 shows the detailed structures of the two networks.

We use subscript t, r to respectively stand for target and reference frame. Here, reference frame is the frame provided with a user-labeled trimap.

4.1. Trimap Propagation

Our trimap propagation method is based on region similarity measures between the reference and target frames. Without computing any optical flow, the network uses two encoders sharing the same structure to respectively extract semantic features of image-trimap $\{I_r, T_r\}$ pair of the reference frame and image I_t of the target frame. We denote the resulting reference feature from the last layer of its corresponding encoder as $\mathcal{F}_r$ and target feature as $\mathcal{F}_t$. To enlarge receptive fields and overcome different motion as well as scales between the reference and target frames, we apply a cross-attention based correlation layer to match the reference and target frames.

This correlation layer is composed of key, queries and memories. Key and queries are used to generate correlation scores between the reference and target frames, while memories are applied to enhance correlated features. Given the features $\mathcal{F}_r$ and $\mathcal{F}_t$ of shape $hw \times c$, we adopt Wang *et al.* [40] to apply a matrix multiplication between queries and keys to get a similarity matrix of shape $hw \times hw$. Intuitively, if a pixel in target frame is highly correlated to a pixel in reference frame, the correlation score between these two pixels will be high. Thus, if a pixel in target frame belongs to the foreground (resp. unknown) region, it should be matched to a corresponding foreground (resp. unknown) pixel in reference frame. These correlation scores are then multiplied with the memory features. The weighted memory features are regarded as residuals and added to $\mathcal{F}_t$. This allows trimap information of reference frames to be propagated to the target frame without affecting the computation of correlation scores. Finally, these aggregated features are decoded to classify all pixels into three categories, i.e., foreground, background or unknown, through a classification head. An example of trimap propagation is provided in Figure 4.

4.2. Encoder-Decoder Network

After generating coarse trimaps for the target frame, our deep video matting framework employs an effective auto encoder-decoder structure to extract features of multiple image-trimap pairs. We first apply an encoder network to extract both low-level structural features and high-level semantic features of pixels. Specifically, this encoder network receives multiple frames with corresponding propagated trimaps to extract pyramid features at different levels. We adopt ResNet-50 [41] as our encoder and collect the features after each residual block. These features obtained at different levels are sent to the decoder for alpha predictions.

In order to predict more accurate alpha matte, a delicate decoder is designed to upscale features from the last layer of the encoder to the same resolution as the input image with several up-convolution layers. We apply sub-pixel convolution layer to upsample features, rather than unpooling operation or deconvolution, for both accuracy and efficiency: unpooling operation generates sparse indices and sometimes leads to zero gradients, while deconvolution suffers efficiency problem.

Our design is adapted from the U-Net architecture [42] with skip-connections to preserve both global context features and local detail information. The main difference of our decoder from other U-Net structures lie in the skip-connections, which are enhanced by a novel spatio-temporal feature aggregation module (ST-FAM).

4.3. Spatio-Temporal Feature Aggregation Module

The main issues in video matting compared to image matting are how to utilize temporal information across multiple frames to help distinguish foreground and background color, and how to improve temporal consistency of alpha mattes. To achieve these goals, we not only need to consider global context information and local detailed structural information in a single frame, but also need to incorporate motion information of moving pixels by utilizing temporal information from neighboring features to enhance our predictions. To this end, we propose a novel spatio-temporal feature aggregation module (ST-FAM) to exploit information inherent in the features at different scales and timestamps. Figure 3 shows the structure of ST-FAM. Overall, it is composed of temporal feature alignment module (TFA) and temporal feature fusion module (TFF). More implementation details of ST-FAM can be found in supplementary materials.

Temporal Feature Alignment Module. The biggest advantage of videos compared to images in matting is that consecutive frames provide temporal information of fore-

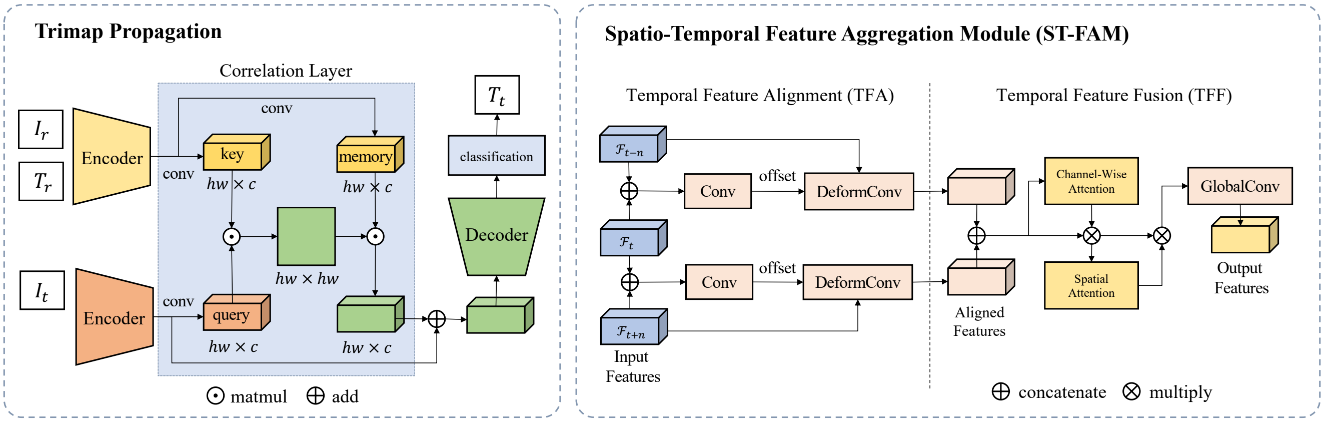


Figure 3. Detailed structure of trimap propagation network and spatio-temporal feature aggregation module (ST-FAM) introduced in Figure 2. In the trimap propagation network, a correlation layer is used to find the mapping of pixels between I_r (reference) and I_t (target); $T_{r|t}$ denotes trimap. ST-FAM is composed of temporal feature alignment (TFA) and temporal feature fusion (TFF), which are respectively responsible for aligning and aggregating features of different frames.

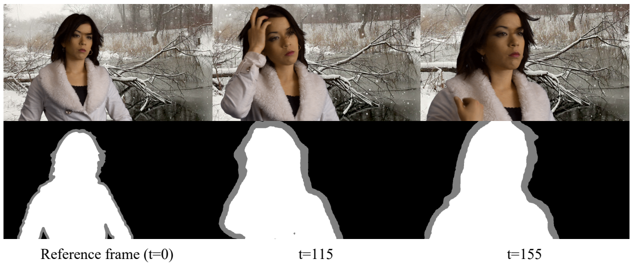


Figure 4. Trimap propagation. The first column is the reference frame and trimap ($t = 0$). The second and third columns are the propagated trimaps at $t = 115$ and $t = 155$ respectively.

ground objects and background scenes. Motion information of pixels is useful in distinguishing foreground and background colors, and can help the model learn more accurate appearance of foreground objects and handle hard matting cases with complex backgrounds. To effectively exploit temporal information, we design a light-weight module that can effectively aggregate temporal features to enhance current expression.

Wang *et al.* [43] proposed a pyramid, cascading and deformable module to deal with motions in video restoration task. Inspired by this work, we make our model aware of motion information by aligning the features of neighboring frames with features of target frame. Specifically, for pixel p at time t of frame I_t, we try to learn offset Δp for p implicitly and translate the offset to obtain aligned features by deformable convolution. Formally, the aligned feature $\mathcal{F}^*$ at time $t + \Delta t$ ($\Delta t \in \{0, \pm 1, \dots\}$) of p is defined as

$$\mathcal{F}^*_{t+\Delta t}(p) = \sum_k w_k \mathcal{F}_{t+\Delta t}(p + \Delta p_k) \tag{2}$$

where k and w_k respectively represent the deformable convolution kernel location and the corresponding weight. Δp_k is a *learnable* offset from the concatenation of features at time t and $t + \Delta t$. Learning offset and aligning features between t and $t + \Delta t$ enable our model to automatically map identical or similar regions and pixels by their high-dimensional feature expression in temporal context, and consequently to encode temporal information within the aligned features.

Temporal Feature Fusion Module. The aligned features obtained by TFA above are passed through the TFF step. As our goal is to obtain fused features for target frame, a simple plausible solution is to compute their average values. However, this will introduce noise or ambiguous information when moving pixels of target frame are lost in neighboring frames. To reduce confusion, our model should pay attention only to relevant information useful for the prediction of target frame. To this end, we introduce an attention mechanism. We perform channel-wise attention as well as spatial attention on the aligned features, by guiding our model to leverage the importance of different channels and the interest regions within a channel. Specifically, we compute a channel attention weight map by applying a global average pooling layer followed with a fully-connected layer on the aligned features, and multiply this map with the aligned features. Then the output feature is multiplied with a learnable spatial attention weight map. Finally, a 1×1 convolution layer and a global convolution layer [44] are applied to reduce channels and enlarge receptive fields respectively.

After this module, we obtain temporally enhanced features from different blocks which are used in the skip connection with the up-scaled features to fully exploit information from both high-level and low-level features. After deriving useful features aggregated in both spatial and temporal dimensions from our decoder, we apply a prediction head, composed of a 3×3 convolution and a sigmoid function, to generate the alpha mattes for target frame.

4.4. Loss Functions

Our network uses multiple losses, including alpha prediction loss and composition loss, which are widely applied in many deep image matting methods. Meanwhile, the gradient loss, KL-divergence loss, and temporal coherence loss are also used.

Alpha Loss. At timestamp t and pixel p, with the alpha matte prediction $\alpha_{p,t}$, and the ground truth $\hat{\alpha}_{p,t}$, we define the difference loss of predicted alpha as

$$L_a = \begin{cases} ||\alpha_{p,t} - \hat{\alpha}_{p,t}||_2 & \text{if } \alpha_{p,t} = 0, 1 \\ ||\alpha_{p,t} - \hat{\alpha}_{p,t}||_1 & \text{otherwise} \end{cases}. \tag{3}$$

We treat the transition region separately from the foreground and background.

Composition Loss. For the composition loss L_c, we calculate L1 loss of the transition region. $\hat{F}$, $\hat{B}$ and $\hat{I}$ denote groundtruth foreground, background and composited frame.

$$L_c = ||\alpha_{p,t}\hat{F}_{p,t} + (1 - \alpha_{p,t})\hat{B}_{p,t} - \hat{I}_{p,t}||_1 \tag{4}$$

Gradient Loss. Let G be the Sobel filter, the gradient loss L_g at timestamp t is defined as

$$L_g = ||G(\alpha_{p,t}) - G(\hat{\alpha}_{p,t})||_1 \cdot L_a \tag{5}$$

Different from treating the absolute difference of gradient as a loss function, we use it as a spatial loss weight, which shares the same idea with online hard example mining.

KL-Divergence Loss. We use KL-divergence loss to provide another constraint for the alpha matte prediction against its ground truth. We first normalize $\alpha_{p,t}$ and $\hat{\alpha}_{p,t}$ by their summation value respectively, and then apply KL-divergence loss, which is defined as

$$L_{kl} = D_{KL}(\frac{\alpha_{p,t}}{\sum \alpha_{p,t}}, \frac{\hat{\alpha}_{p,t}}{\sum \hat{\alpha}_{p,t}}) \tag{6}$$

where D_{KL} represents the KL-divergence function.

Temporal Coherence Loss. We enforce consistency of the predicted alpha values between consecutive frames by defining the temporal coherence loss at timestamp t as

$$L_{tc} = ||\frac{\mathrm{d}\alpha_{p,t}}{\mathrm{d}t} - \frac{\mathrm{d}\hat{\alpha}_{p,t}}{\mathrm{d}t}||_2. \tag{7}$$

Total loss. Finally, the total loss is the summation of all pixels at all target timestamps, defined as

$$L_{total} = \frac{1}{\#}\sum_{p,t}(L_a + L_c + L_g + L_{kl} + L_{tc}) \tag{8}$$

where $\#$ denotes the number of pixels.

5. Experiments

5.1. Implementation Details

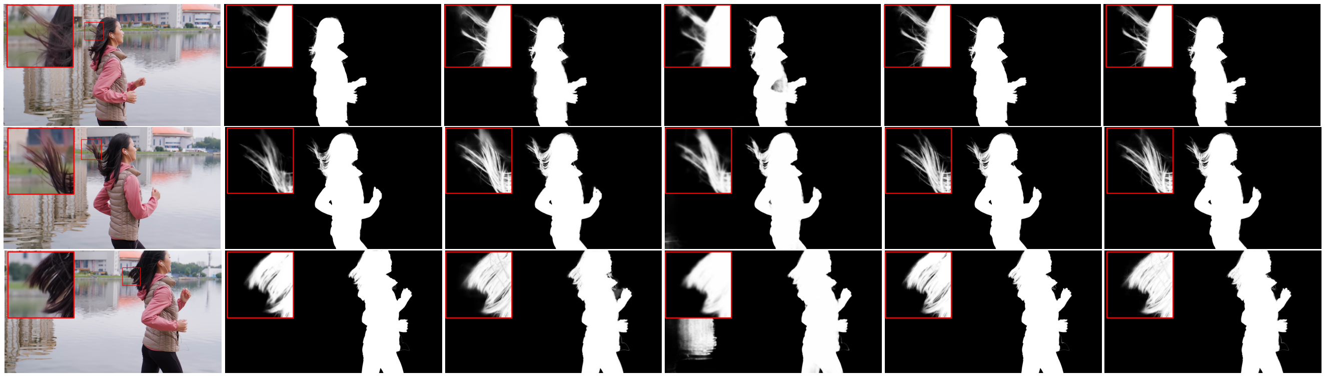
Trimap Propagation. Our trimap propagation network applies two encoders adapted from ResNet-34 [41] and an decoder composed of by several up-sampling and convolution layers. In the training stage, two frames are randomly sampled from a video. One is treated as the reference and the other as target. Augmentations including random cropping, coloring jittering and flipping are performed on two frames. We totally train the model for 75 epochs with a batch size of 4. The initial learning rate is set to 0.001 and then decays linearly to an end learning rate of 0.0001. Adam optimizer is applied for training all the parameters.

Encoder-Decoder Network. In the training stage, for each sample, we randomly select a chunk of continuous frames from the whole video as the input frames. We treat the middle frame as the target frame, and the others as neighboring frames. Then we randomly pick a 320×320 patch centered on pixels in the unknown regions of the target frame and crop the $320 \times 320 \times (2n + 1)$ cube from the chunk where n is the number of neighboring frames. To make the model robust to scale variance, we crop cubes with different sizes including 320×320, 480×480, 640×640, and resize them to 320×320. Then, we apply random horizontal flipping on the cube. The above operations are conducted consistently on the composited frames and their alpha, foreground and background frames. In addition, the trimaps for the cube are randomly generated from the groundtruth alpha mattes. We dilate and erode the ground truth alpha mattes with a random kernel size within a range of $[2, 5]$ and a random iteration within range of $[5, 15]$.

We initialize our encoder network with the pre-trained weights on the ImageNet [45] dataset and the fourth input channel with zeros. The decoder network is initialized with Xavier random variables. All models are trained for 100 epochs with batch size of 1 and n set to 2. The initial learning rate is 0.00005 which is fixed in the first 20 epochs and decays linearly in the last 80 epochs with a decay rate of 0.98. We use Adam optimizer to update parameters for the whole network.

5.2. Evaluation Metrics

We use both image-based and video-based evaluation metrics. To evaluate the per-pixel accuracy, we follow Xu *et al.* [6] and adopt four quantitative metrics, namely the sum of absolute differences (SAD), mean square error (MSE), the gradient error (Grad) and the connectivity error (Conn). In addition, to evaluate the temporal coherency, we also take dtSSD and MESSDdt into consideration. These two metrics were proposed in [38] and defined as



Frame GCA Context-Aware LateFusion Ours(full-trimap) Ours(20-frame)

Figure 5. Comparison of alpha predictions with image-based methods on real-world high-resolution videos. GCA [46] and Context-Aware [47] take frame-by-frame user-supplied trimaps as inputs. Our 20-frame model takes the trimaps propagated from the nearest reference frames provided every 20 frames.

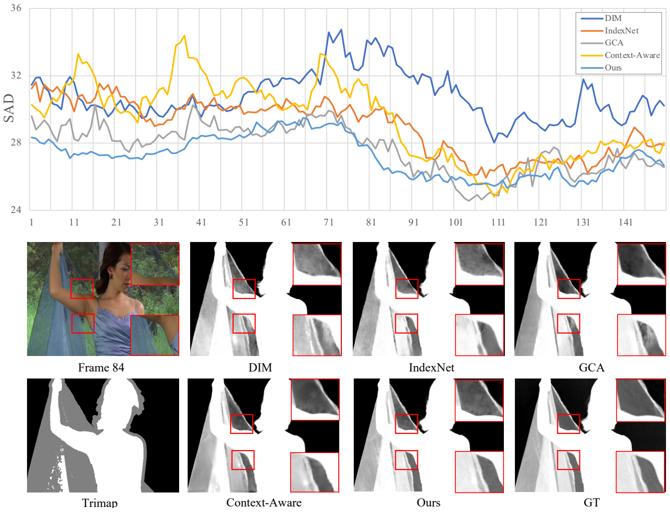


Figure 6. Comparisons with image-based methods on metric SAD.

$$\mathrm{dtSSD} = \frac{1}{\#}\sum_{t}\sqrt{\sum_{p}(\frac{d\alpha_{p,t}}{dt} - \frac{d\hat{\alpha}_{p,t}}{dt})^2} \tag{9}$$

$$\begin{aligned} \mathrm{MESSDdt} =& \frac{1}{\#}\sum_{p,t}|(\alpha_{p,t} - \hat{\alpha}_{p,t})^2 - \\ & (\alpha_{p+v_p,t+1} - \hat{\alpha}_{p+v_p,t+1})^2| \end{aligned} \tag{10}$$

Here v_p denotes the motion vector at pixel p, which is computed by optical flow algorithm for groundtruth sequences. The evaluation code of the two temporal metrics are our own implementations.

5.3. Results on Composited Dataset

We evaluate our method and image-based methods on the proposed composited test set under different trimap settings, including “full-trimap” and “20-frame”, in which user-labeled trimaps are respectively provided frame-by-frame and every 20 frames. Table 1 tabulates evaluation results, where our model outperforms image-based methods on all of the metrics by a large margin under dense-trimap setting. Even under the 20-frame setting, our method still achieves state-of-the-art performance. In addition, we also plot the quantitative results of a sample on metric SAD frame-by-frame in Figure 6. Compared to image-based methods, our model generates more accurate and consistent alpha mattes.

5.4. Results on Real-world High-Resolution Videos

While experiments on our composited test set have demonstrated the our model’s effectiveness, a robust matting method should generalize well to real-world videos. Figure 5 compares the alpha mattes of our method and image-based methods on real-world videos, showing that better foreground mattes can be extracted with proper consideration of the temporal information than the image-based methods directly applied to videos. See supplementary materials for more comparisons and composition results.

6. Ablation Studies

6.1. Effects of TFA and TFF

ST-FAM is composed of TFA and TFF. TFA employs deformable convolution to align features of neighboring frames. TFF applies channel-wise and spatial attention to aggregate temporal features. In order to justify the effectiveness of this design, we train a basic model without ST-FAM and then add our TFA as well as TFF in turn. Table 2 shows the comparison results. Compared to the basic model in the first row, our TFA outperforms by 1.94. With our novel TFF module, the performance is further promoted by 1.41. TFA and TFF modules benefit our model in aligning temporal features and discarding harmful neighboring information, which is conducive to the effective extraction of foreground objects with complex backgrounds.

Methods	Trimap Setting	SAD	MSE	Grad	Conn	dtSSD	MESSDdt
DIM [6]	full-trimap	54.55	0.030	35.38	55.16	23.48	0.53
IndexNet [48]	full-trimap	53.68	0.028	27.52	54.44	19.50	0.49
Context-Aware [47]	full-trimap	51.78	0.027	28.57	49.46	19.37	0.50
GCA [46]	full-trimap	47.49	0.022	26.37	45.23	18.36	0.33
Ours	full-trimap	**40.91**	**0.014**	**19.02**	**40.58**	**15.11**	**0.25**
LateFusion [26]	no-trimap	69.62	0.042	45.34	70.70	38.59	0.71
Ours	20-frame	43.66	0.016	26.39	42.23	16.34	0.28

Table 1. Results of our deep video matting versus image-based matting methods on the composited test set. "full-trimap" means frame-by-frame trimaps are provided; "no-trimap" does not use trimaps. Under "N-frame" setting, trimaps are provided for every Nth frame.

Method	SAD	MSE	dtSSD
basic	44.26	0.016	16.57
basic + TFA	42.32	0.015	15.69
basic + TFA + TFF	40.91	0.014	15.11

Table 2. Results of TFA and TFF.

n	1	2	3	4
SAD	48.72	46.98	46.29	46.30
dtSSD	19.24	18.59	18.17	18.10

Table 3. Results of temporal aggregation.

Method	SAD	MSE	dtSSD
naive-fusion	44.13	0.016	16.62
cross-attention-fusion	41.29	0.015	15.37
flow-fusion	44.06	0.016	16.34
ST-FAM (Ours)	**40.91**	**0.014**	**15.11**

Table 4. Results of different temporal fusion networks.

Trimap Setting	SAD	MSE	dtSSD
full-trimap	40.91	0.014	15.11
20-frame	43.66	0.016	16.34
40-frame	52.85	0.026	19.23
1-trimap	65.33	0.039	35.46

Table 5. Results of different trimap settings.

6.2. Effects of Temporal Aggregation

Temporal information from neighboring frames help the model distinguish foreground and background pixels. The context information along temporal dimension also has an impact on the aggregation, which is decided by the number of neighboring frames in our framework. To exploit ST-FAM's ability of aggregating temporal information, we conduct experiments on models with different n neighboring frames. Table 3 shows that when we increase n before the saturation point, the model learns temporal information from more adjacent frames so that it has a better understanding of object's motion and give more accurate and consistent predictions of alpha mattes.

6.3. Effects of Temporal Fusion Network

Making use multiple frames constitutes the main difference between video-based methods and image-based methods. To exploit an effective design of aggregating temporal information in matting task, we compare several temporal fusion networks, including naive-fusion, cross-attention-fusion, and flow-fusion. Naive-fusion aggregates temporal information through several 3×3 convolution layers, while cross-attention-fusion applies a cross-attention based correlation layer to compute the similarities between two frames. Flow-fusion obtains the motion vector of pixels via a lightweight flow estimation network, and the estimated motion vector is concatenated with features from the decoder for final predictions. More implementation details are provided in supplementary materials.

Table 4 shows the quantitative comparisons. Although the cross-attention-fusion is also capable of aggregating temporal information, the cost of computation increases rapidly when more neighboring frames are integrated.

6.4. Effects of Trimap Propagation

To evaluate the performance of our trimap propagation method, we compare our matting results under different trimap settings: full-trimap mode, N-frame mode ($N > 1$), and 1-trimap mode. In the full-trimap mode frame-by-frame trimaps are provided by users; in the N-frame mode user-supplied trimaps are provided for every Nth frame; in 1-trimap mode only one user-supplied trimap at $t = 0$ is provided for the entire video. For each target frame, its nearest neighbor is found in temporal domain that contains the user-supplied trimap as the reference frame.

Table 5 tabulates the quantitative results of the different trimap settings. We can see that even under the 20-frame setting our performance only drops slightly. Qualitative results can be found in supplementary materials.

7. Conclusion

This paper proposes a new deep video matting framework that exploits temporal information between the target and reference as well as neighboring frames. This framework consists of an encoder-decoder structure using novel spatio-temporal feature aggregation modules. The proposed module benefits our model in enhancing temporal coherence leading to significantly better alpha prediction in objects with rapid motions or complex backgrounds. This paper also contributes a large-scale video matting dataset that covers a great variety of unique matting cases to complete the data gap in present and future deep video matting research. We have conducted extensive experiments on our proposed test set and real-world high-resolution videos to validate our method on dealing with complex scenes.

References

[1] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 1

[2] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1

[3] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *CVPR*, 2020. 1

[4] Inchang Choi, Minhaeng Lee, and Yu-Wing Tai. Video matting using multi-frame nonlocal matting laplacian. In *ECCV*, 2012. 1, 2, 3

[5] Dongqing Zou, Xiaowu Chen, Guangying Cao, and Xiaogang Wang. Unsupervised video matting via sparse and low-rank representation. *TPAMI*, 42(6):1501–1514, 2019. 1, 2, 3

[6] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. 2017. 1, 2, 3, 6, 8

[7] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *CVPR*, 2001. 2

[8] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. A cluster sampling method for image matting via sparse coding. In *ECCV*, 2016. 2

[9] Eduardo SL Gastal and Manuel M Oliveira. Shared sampling for real-time alpha matting. *Computer Graphics Forum*, 29(2):575–584, 2010. 2

[10] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR*, 2011. 2

[11] Mark A Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *CVPR*, 2000. 2

[12] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *CVPR*, 2017. 2

[13] Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics*, 37(4):1–13, 2018. 2

[14] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007. 2

[15] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. In *CVPR*, 2012. 2

[16] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, 2005. 2

[17] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. In *CVPR*, 2006. 2

[18] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. In *CVPR*, 2007. 2

[19] Jue Wang, Michael F Cohen, et al. Image and video matting: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(2):97–175, 2007. 2

[20] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *ECCV*, 2016. 2

[21] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. In *BMVC*, 2018. 2

[22] Yu Wang, Yi Niu, Peiyong Duan, Jianwei Lin, and Yuanjie Zheng. Deep propagation based image matting. In *IJCAI*, 2018. 2

[23] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *ECCV*, 2016. 2

[24] Bingke Zhu, Yingying Chen, Jinqiao Wang, Si Liu, Bo Zhang, and Ming Tang. Fast deep matting for portrait animation on mobile phone. In *ACM MM*, 2017. 2

[25] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *ACM MM*, 2018. 2

[26] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion cnn for digital matting. In *CVPR*, 2019. 2, 8

[27] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian-sheng Hua. Boosting semantic human matting with coarse annotations. In *CVPR*, 2020. 2

[28] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. Attention-guided hierarchical structure aggregation for image matting. In *CVPR*, 2020. 2

[29] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *CVPR*, 2020. 2

[30] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Transactions on Graphics*, 21(3):243–248, 2002. 2, 3

[31] Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. Temporally coherent video matting. In *SIGGRAPH*, 2010. 2, 3

[32] Jue Wang and Michael F. Cohen. Optimized color sampling for robust matting. 2007. 2

[33] Xue Bai, Jue Wang, and David Simons. Towards temporally-coherent video matting. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, 2011. 2, 3

[34] Dingzeyu Li, Qifeng Chen, and Chi-Keung Tang. Motion-aware knn laplacian for video matting. In *ICCV*, 2013. 2, 3

[35] Neel Joshi, Wojciech Matusik, and Shai Avidan. Natural video matting using camera arrays. *ACM Transactions on Graphics*, 25(3):779–786, 2006. 2

[36] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F Hughes, and Frédo Durand. Defocus video matting. *ACM Transactions on Graphics*, 24(3):567–576, 2005. 2

[37] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *CVPR*, 2009. 3

[38] Mikhail Erofeev, Yury Gitman, Dmitriy Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, 2015. 3, 6

[39] Keylight. https://learn.foundry.com/nuke/content/reference_guide/keyer_nodes/keylight.html. 3

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 6

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 4

[43] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019. 5

[44] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters - improve semantic segmentation by global convolutional network. In *CVPR*, 2017. 5

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[46] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *AAAI*, 2020. 7, 8

[47] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *ICCV*, 2019. 7, 8

[48] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In *ICCV*, 2019. 8