Attention-guided Temporally Coherent Video Object Matting

Yunke Zhang
Zhejiang University
yunkezhang@zju.edu.cn

Chi Wang
Zhejiang University
wangchi1995@zju.edu.cn

Miaomiao Cui
Alibaba Group
miaomiao.cmm@alibaba-inc.com

Peiran Ren
Alibaba Group
peiran.rpr@alibaba-inc.com

Xuansong Xie
Alibaba Group
xingtong.xxs@taobao.com

Xian-Sheng Hua
Damo Academy, Alibaba Group
xiansheng.hxs@alibaba-inc.com

Hujun Bao
Zhejiang University
bao@cad.zju.edu.cn

Qixing Huang
The University of Texas at Austin
huangqx@cs.utexas.edu

Weiwei Xu*
Zhejiang University
xww@cad.zju.edu.cn

ABSTRACT

This paper proposes a novel deep learning-based video object matting method that can achieve temporally coherent matting results. Its key component is an attention-based temporal aggregation module that maximizes image matting networks' strength for video matting networks. This module computes temporal correlations for pixels adjacent to each other along the time axis in feature space, which is robust against motion noises. We also design a novel loss term to train the attention weights, which drastically boosts the video matting performance. Besides, we show how to effectively solve the trimap generation problem by fine-tuning a state-of-the-art video object segmentation network with a sparse set of user-annotated keyframes. To facilitate video matting and trimap generation networks' training, we construct a large-scale video matting dataset with 80 training and 28 validation foreground video clips with ground-truth alpha mattes. Experimental results show that our method can generate high-quality alpha mattes for various videos featuring appearance change, occlusion, and fast motion. Our code and dataset can be found at: https://github.com/yunkezhang/TCVOM

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; *Video segmentation*; **Neural networks**; **Supervised learning**.

KEYWORDS

datasets, neural networks, video matting, attention mechanism

ACM Reference Format:
Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, Qixing Huang, and Weiwei Xu. 2021. Attention-guided Temporally Coherent Video Object Matting. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475623

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475623

1 INTRODUCTION

The task of video object matting is to compute temporally coherent alpha mattes for a foreground video object at each frame. It is a fundamental task for many video editing applications, *e.g.* compositing the foreground object into new background videos. The resulting alpha mattes represent the fractional opacity (between 0 and 1) of pixels. Such opacity mainly comes from the transparency or the partial coverage of background pixels around the foreground object boundaries. Specifically, the matting problem tries to solve for three types of unknowns at each pixel, i.e., the foreground color F, the background color B, and the alpha value α, based on the measured pixel color C, where $C = \alpha F + (1 - \alpha)B$. Moreover, to facilitate image and video matting, a trimap [58] is usually required to separate an image into the foreground region (FR), the background region (BR), and the unknown region (UR). Here UR covers partial or transparent foreground object boundaries.

Video object matting is related to image matting in the sense that each frame of the matting output essentially solves the corresponding image matting problem. The matting problem is challenging since the number of unknowns exceeds the number of measured colors. Thus, it is critical to build priors to constrain the solution space [1, 11, 15, 32]. State-of-the-art (SOTA) image matting algorithms typically build on convolutional neural network (CNN). They improve the image matting results significantly by learning multi-scale features to predict alpha values for pixels in the UR [7, 10, 12, 24, 39, 54, 61]. Given an input video clip and its corresponding trimap for each frame, one can perform video matting with any image matting method by processing each video frame independently. However, this approach may lead to temporal incoherence in the obtained alpha mattes (*e.g.* flickering, shown in the third row of Figure 1). To improve temporal coherence, existing video matting methods exploit temporal correspondence between video frames, such as optical flow, to construct multi-frame alpha or color priors or compute temporal affinities to incorporate motion cues [3, 13, 14, 34, 74]. However, they rely on local color distributions as main features and may suffer from motion ambiguities at transparent pixels, resulting in flickering or blocky artifacts in the matting results.

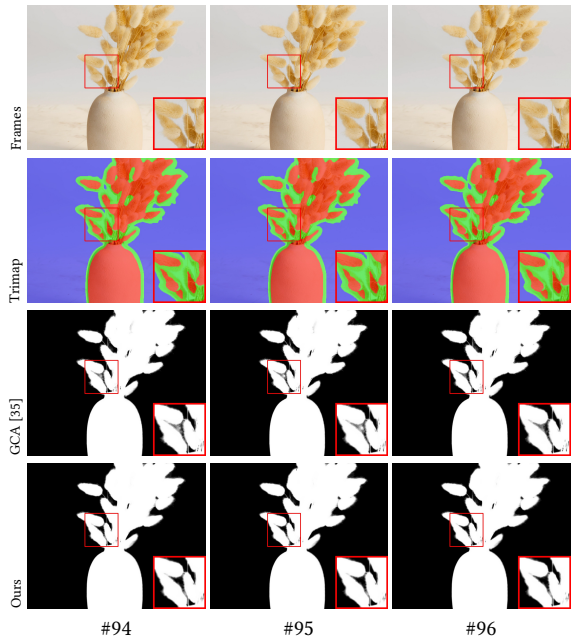

Figure 1: A video matting result comparison using an Internet video clip "plant". Trimaps are generated using our trimap generation method. Red, blue and green color correspond to FR, BR, and UR respectively. "#" denotes the frame number. Our method is capable of generating more temporally coherent result compared to GCA [35], an image matting network. Please see the supplementary video for the complete result.

This paper proposes a novel CNN-based video object matting method to achieve temporally coherent results. Its essential component is a simple yet effective attention-based temporal aggregation module (TAM) that can be seamlessly combined with SOTA image matting networks, such as GCA [35], IndexNet [38] (Index) and DIM [61], extending them into video matting networks. This simple design maximizes image matting networks' strength and yields a substantial performance boost for video matting, especially on temporal-related metrics. We leverage the widely used attention mechanism to compute the temporal attention weights [56, 59] for a pair of pixels adjacent to each other along the time axis. Conceptually, these weights are analogous to the non-local, temporal affinity values used in traditional affinity-based video matting methods [13, 17]. However, the attention weights are computed using high-dimensional features rather than local color and motion features. Moreover, we design a novel target affinity term to supervise the learning of attention weights. This term's ground-truth is automatically derived from the alpha matte and used in a cross-entropy loss to guide the training. Such design significantly improves our method's robustness against noises due to video compression, appearance change and motion. As shown in Figure 1, our method (the last row) can generate much more temporally coherent result.

Another challenge is generating trimaps for an input video clip to fulfill the task of video object matting. To this end, we propose to train the space-time memory network (STM) [42], which is a semi-supervised video object segmentation (VOS) network, to segment each frame into FR, BR and UR. It only requires the user to annotate trimaps of a target object at several keyframes, usually three to five frames for a video clip of around 200 frames in our experiments, which enhances the efficiency of video matting significantly. To handle the large variations of user-annotated keyframe trimaps, we perform online-finetuning on the STM network. We then ensemble the bidirectional prediction results to improve the quality of generated trimaps.

In summary, the main contributions of this work are:

- We propose a temporal aggregation module that integrates image matting networks to achieve temporally coherent video matting results. It leverages the attention mechanism to compute temporal affinity values in the feature space, resulting in a robust matting method to handle challenging videos featuring appearance change, occlusion, and fast motion.
- We propose an STM-based trimap generation method to enhance the efficiency of video matting greatly. The user only needs to annotate trimaps at several keyframes to generate trimap for every video frame.
- To enable video object matting and trimap generation networks training, we construct a video object matting dataset, termed VideoMatting108, that covers various objects and different types of motions. In total, our dataset has 108 foreground video clips with ground-truth alpha mattes, all in 1080p resolution, averaging in 821 frames per clip. The dataset will be made publicly available.

2 RELATED WORKS

Image matting. The sampling-based image matting methods [15, 19, 21, 23, 46] build the FR and BR color priors using the sampled pixels to infer the alpha values, while the affinity-based methods [1, 2, 4, 11, 22, 32, 33, 51] propagate the alpha values from the known FR and BR pixels to the UR pixels based on affinity score and have proven to be robust when dealing with complex images [15, 21, 46]. The deep learning-based matting methods usually train a convolutional encoder-decoder neural network to predict alpha values or foreground/background colors with user-specified trimaps [7, 10, 12, 24, 35, 38, 39, 54, 72]. Recently, "trimap-free" image matting methods also received much attention as they do not require user annotation. Some of the methods use other forms of prior instead of trimaps, *e.g.* background image [47], rough segmentation map or coarse alpha matte [67]. Others do not use any prior at all [37, 45, 70]. The most used image matting dataset in this line of research is provided by Xu *et al.* [61], and a larger dataset is proposed recently by Qiao *et al.* [45].

Video matting. The central problem of video matting is how to obtain temporally coherent alpha mattes. Chuang *et al.* [14] proposed to interpolate manually specified trimaps at key-frames using optical flow, estimate the background pixels, and then perform Bayesian matting at each frame with the estimated background. Motion cues and prior distributions for alpha values and multi-frame colors are widely used in video matting [3, 48, 49, 60]. In [13, 17, 31, 34], spatio-temporal edges between pixels are constructed to compute the alpha mattes for video frames simultaneously. These methods are the extension of affinity-based methods to video matting, which is time-consuming due to the Laplacian matrix's fast-growing size. Zou *et al.* [74] proposed to select nonlocal neighbors through sparse

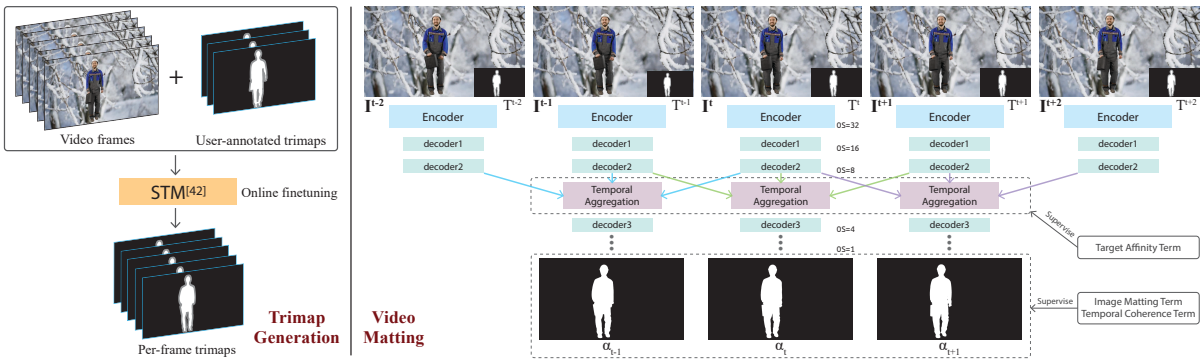

Figure 2: The flowchart of our method during training. "OS" denotes output stride. We do not show encoder-decoder skip connections for clarity. All networks and modules share the same weight across different frames.

coding to constrain pixels having similar features in different frames to get similar alpha values. Besides, depth information can help to construct trimaps and differentiate between pixels of similar colors [73]. Hardware-assisted methods in [26, 40] automatically generate and propagate trimaps in all video frames and optimize for high-quality alpha mattes. Recently, CNN-based video matting methods gained much attention. Lin *et al.* [36] and Ke *et al.* [27] proposed real-time CNN-based methods for trimap-less human portrait matting. However, both of the methods do not enforce the temporal consistency between video frames during training, which may lead to temporally incoherent results. Our method on the other hand, is a trimap-based video matting method that can handle different types of objects with explicitly supervised temporal consistency. Concurrent to our work, Sun *et al.* [52] also proposed a CNN-based video matting method that focuses on temporal coherency.

Attention mechanisms in segmentation and matting. Attention mechanism provides an effective way for neural networks to reinforce correlated features and suppress feature noise, leading to a performance boost in segmentation. There are two main variations of this mechanism. One is the channel-wise self-attention pioneered by Hu *et al.* [25]. Given an input feature tensor, it leverages the global average pooling and the fully-connected layer to infer a channel-wise weight vector to modulate feature maps. The other is the non-local block proposed by Wang *et al.* [59]. It computes the spatiotemporal correlation as the attention, reinforcing the consistency of feature maps effectively. The channel-wise attention approach is widely adopted in image segmentation [65, 66, 69]. Many methods [20, 63, 64, 68, 71] exploit variants of non-local attention modules to capture the spatial long-range dependency. For image matting tasks, the attention mechanism is mostly used for fusing high and low-level image features. Qiao *et al.* [45] adopted both channel-wise and spatial attention for trimap-free matting since high-level image features are the key to recognize a foreground object. GCA [35] also utilizes high-level image features as the attention map to guide the low-level alpha features, achieving SOTA performance in image matting. We thus employ GCA as one of the base matting network structures. Several recent VOS methods also utilize the attention mechanism to fuse features from different video frames for improving temporal consistency [41, 57]. Oh *et al.* [42] extended the memory network approach used in NLP to VOS, which is also a variation of the spatiotemporal attention mechanism. Yang *et al.* [62] extended this idea by matching both the foreground and background with multi-scale features in those frames, achieving SOTA performance. Our method also leverages the attention mechanism for temporally coherent matting. Nevertheless, our attention module is bi-directional, and we use additional temporal loss terms to supervise the network training.

Temporal coherence. One standard solution to temporal coherence is the temporal smoothing filter, which considers the spatial and temporal adjacent pixels simultaneously [8, 9, 30, 43]. Another solution is to impose the temporal coherence in the post-processing, which is blind to image filters [5, 29]. In contrast, our method does not rely on temporal smoothing filter but the feature-space affinity to produce temporally coherent alpha mattes.

3 OUR METHOD

Given an input video, our method first runs trimap generation to propagate the user-annotated trimaps to the other frames. We then run a video matting network, formed by integrating temporal aggregation module (TAM) into an image-based matting network, to obtain a temporally coherent alpha matte at each frame (See Figure 2). When computing an alpha matte for frame $\mathbf{I}^t$ in testing stage, TAM only needs to aggregate the CNN features from three consecutive frames, *i.e.* $\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1}$. The choice of three consecutive frames offers great flexibility in network design while ensures computational efficiency. However, during training, our network takes five consecutive frames simultaneously as inputs, *i.e.* $\mathbf{I}^{t-2}, ..., \mathbf{I}^{t+2}$, and predicts $\alpha^{t-1}, \alpha^t, \alpha^{t+1}$ to facilitate the computation of loss functions. Note that we choose to integrate TAM into the base network at the decoder stage of output stride (OS) 8. It indicates that the resolution of the feature map should be $H/8 \times W/8$, where H, W is the input image resolution. This choice is to balance computational cost and feature level, and we empirically found that OS=8 is a good trade-off (see the supplementary material for the OS experiment).

In the following, we will first describe the design of TAM (Sec. 3.1), and proceed to describe the training loss (Sec. 3.2) and the training strategy of TAM (Sec. 3.3). Finally, we describe the details of trimap generation using STM [42] (Sec. 3.4) and our video object matting dataset (Sec. 3.5).

3.1 Temporal Aggregation Module

Figure 3 illustrates the structure of TAM. It leverages the attention mechanism to aggregate features from $\mathbf{I}^{t-1}$ and $\mathbf{I}^{t+1}$ with features

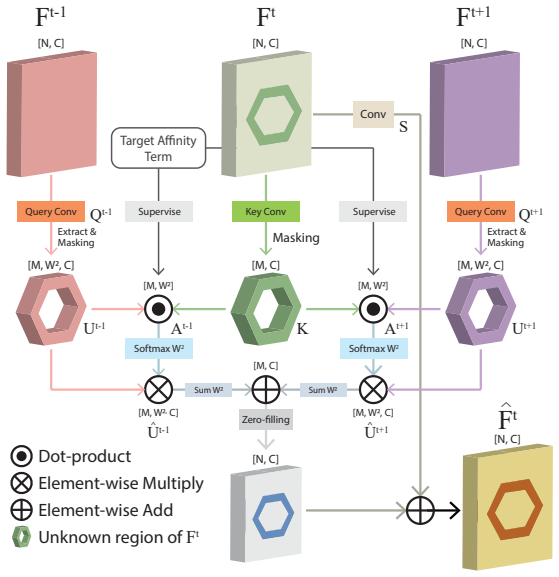

Figure 3: The architecture of our temporal aggregation module.

from $\mathbf{I}^t$ for pixels inside the UR of $\mathbf{I}^t$. This design benefits temporal coherence by encoding temporal correlation in the aggregated features at frame $\mathbf{I}^t$.

For a video frame $\mathbf{I}^t$, we denote its input feature map as $\mathbf{F}^t \in \mathbb{R}^{N\times C}$. Here C is the total number of channels, and N is the total number of pixels in this feature map. TAM takes three feature maps $\mathbf{F}^{t-1}, \mathbf{F}^t, \mathbf{F}^{t+1}$ as inputs. First, we compute key and query features with separate convolution layers. Specifically, we feed $\mathbf{F}^{t-1}, \mathbf{F}^{t+1}$ into a 3×3 query convolution layer. With $\mathbf{Q}^{t-1}, \mathbf{Q}^{t+1}$ we denote the output query feature map. In contrast, we input $\mathbf{F}^t$ into a 3×3 key convolution layer and denote the output feature map as $\mathbf{K}$.

Since we only focus on UR of $\mathbf{I}^t$ in TAM, we extract the UR in $\mathbf{K}$ using a mask generated by down-sampling the original trimap to the same resolution. For each pixel i within the UR of $\mathbf{K}$, we extract two $W\times W$ feature patches centered at the same position of pixel i from features $\mathbf{Q}^{t-1}$ and $\mathbf{Q}^{t+1}$, respectively. We denote this window as $\mathbb{W}$ and the extracted patch features as $\mathbf{U}^{t-1}, \mathbf{U}^{t+1} \in \mathbb{R}^{M\times W^2\times C}$, where M is the number of pixels in the UR.

Next, we compute the attention weights between $\mathbf{K}$ and $\mathbf{U}^{t-1}$ and similarly for $\mathbf{K}$ and $\mathbf{U}^{t+1}$, and then use the attention weights as temporal affinity values to modulate the corresponding adjacent frame features. Taking $\mathbf{K}$ and the query features $\mathbf{U}^{t-1}$ of $\mathbf{I}^{t-1}$ as an example, we compute the set of affinity values $\mathbf{A}^{t-1} \in \mathbb{R}^{M\times W^2}$ as follows:

$$\mathbf{A}^{t-1}(i,j) = \frac{e^{\mathbf{K}(i)\cdot\mathbf{U}^{t-1}(i,j)}}{\sum_{k=1}^{W^2} e^{\mathbf{K}(i)\cdot\mathbf{U}^{t-1}(i,k)}}, i \in UR, j \in \mathbb{W}. \quad (1)$$

where $\mathbf{A}^{t-1}(i,j)$ denotes the affinity value computed for a pixel i of $\mathbf{I}^t$ and a pixel j within the patch at $\mathbf{I}^{t-1}$; note that $\cdot$ denotes the point-wise dot product. This operation effectively matches the key feature $\mathbf{K}(i)$ of pixel i with features from $\mathbf{U}^{t-1}$ inside a corresponding $W\times W$ local patch. While our module only uses a local patch instead of the full feature map in the computation attention weights, empirically, it is enough to capture the temporal correlation since the motion between adjacent frames in a video sequence is relatively small. The patch size is set to 7×7 throughout our experiments if not noted otherwise. Note that we use feature maps of OS=8, which can cover a broad range of motion.

Finally, we formulate feature aggregation for $\mathbf{I}^t$ as follows:

$$\begin{aligned}\hat{\mathbf{F}}^t(i) &= \begin{cases} \mathbf{S}(i) + \hat{\mathbf{U}}^{t-1}(i) + \hat{\mathbf{U}}^{t+1}(i) & , i \in UR \\ \mathbf{S}(i) & , Otherwise \end{cases} \\ \hat{\mathbf{U}}^f(i) &= \sum_{j=1}^{W^2} \mathbf{A}^f(i,j)\mathbf{U}^f(i,j), f = \{t-1, t+1\}\end{aligned} \quad (2)$$

where the modulated feature $\hat{\mathbf{U}}^{t-1} \in \mathbb{R}^{M\times C}$ is a weighted sum of all features inside the local patch using $\mathbf{A}^{t-1}$, and $\mathbf{S}$ indicates the features output by a 3×3 convolution layer at frame $\mathbf{I}^t$. The features $\hat{\mathbf{F}}^t$ are fed into next convolution layer to continue the workflow of the original network.

Note that the TAM's design can be seen as sharing the weight between the "query" and "value" convolution in the conventional "KQV" structure. We choose this design because it achieves the best performance among other weight sharing configurations. The influence of the weight sharing will be investigated later in Sec. 4's ablation study.

3.2 Loss function

The overall loss function is a composition of three different terms: the image matting term L_{im}, the temporal coherence term L_{tc}, and the target affinity term L_{af}. In the following, we describe these three terms in details.

Image matting term. The image matting term L_{im} is directly inherited from the deep learning-based image matting methods. This term only considers the single frame prediction results w.r.t. its corresponding ground truth. Common choices of this term are the combination of L_1 alpha matte loss, L_1 alpha gradient loss, composition loss [61] and Laplacian loss [64]. In all our experiments, we set L_{im} the same as the image matting method we chose to use as the base network. Hence, for DIM [61] and Index [38], we use alpha matte loss, alpha gradient loss, and composition loss; for GCA [35], we only use alpha matte loss.

Temporal coherence term. We leverage the temporal coherence metric proposed by Erofeev *et al.* [18] as the temporal coherence term L_{tc}, which can be expressed as follows:

$$L_{tc}^{p,q}(i) = |(\alpha_i^{t-1} - \alpha_i^t) - (\hat{\alpha}_i^{t-1} - \hat{\alpha}_i^t)|_1, \quad (3)$$

where $\alpha, \hat{\alpha}$ are the predicted and ground-truth alpha matte. i denotes a pixel in the UR of frame p. This term is denoted by "dtSSD" in [18], which penalizes the temporal gradient of alpha mattes in our work. There is another temporal coherence metric, termed as "MESSDdt" in [18], which augments Eq. 3 with motion vectors for the purpose of better correspondence between frame $t-1$ and frame t. However, we found that even computed with SOTA optical flow algorithms [55], there are severe motion noises for translucent pixels, which hurts the performance of the network when using "MESSDdt" as the temporal coherence loss. Thus, we do not use "MESSDdt" in our implementation. The quantitative result of using "MESSDdt" as a loss term can be found in the supplementary material.

Target affinity term. While the attention mechanism works well in many cases, Yu *et al.* [64] demonstrates that providing direct supervision for the attention weights can further boost the performance. Inspired by this work, we thus design the target affinity term L_{af}. For a pixel $i \in$ UR at $\mathbf{I}^t$, the target probability of having a

large attention weight between i and a pixel $j \in \mathbb{W}$ inside its local patch at a neighboring frame f can be formulated as:

$$G^f(i,j) = \begin{cases} 1-s, & |\hat{\alpha}_i^t - \hat{\alpha}_j^f| < \theta \\ 0, & Otherwise \end{cases}, \quad (4)$$

where $f = t-1$ or $t+1$, and θ is set to be 0.3. In addition, we follow the label smoothing technique [53] to avoid over-confident decision by introducing the parameter $s = 0.2$. The target probability is computed according to the ground-truth alpha mattes. The goal of this loss term is to make the network learn to assign small affinity values to pixels with large alpha value differences. Therefore, we model the target affinity term between pixel i and j as

$$L_{af}^f(i,j) = \mathbf{BCE}(\Phi(\mathbf{K}(i) \cdot \mathbf{U}^f(i,j)), G^f(i,j)) \quad (5)$$

where **BCE** denotes the binary cross-entropy function, and Φ denotes the sigmoid function. During training, the term L_{af} is calculated as $L_{af} = \frac{1}{2}(L_{af}^{t-1} + L_{af}^{t+1})$.

In summary, our network is trained by the weighted average of these three terms:

$$L = w_{im}L_{im} + w_{tc}L_{tc} + w_{af}L_{af}. \quad (6)$$

where we set $w_{im} = 1$, $w_{tc} = 0.5$, and $w_{af} = 0.25$ in our experiments.

3.3 Training strategy

The training of our video matting network consists a pre-training stage and a main stage. In the following, we denote the video matting network as GCA+TAM, DIM+TAM or Index+TAM which corresponds to base methods GCA [35], DIM [61] and Index [38] respectively. For pre-training, we input three frames $\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1}$ along with trimaps and predict the center frame alpha matte α^t using the supervision from L_{im} only. During pre-training, all layers before the TAM in the network are initialized and fixed using the pre-trained weight of a off-the-shelf image matting network. TAM and the rest of the decoder layers are randomly initialized and trained on the DIM dataset [61]. The dataset is augmented with random affine transformations (rotation, translation, and scaling) to generate sequences of three frames. Random flipping and cropping are also conducted for further augmentation. We pre-trained the network for 20 epochs using an input resolution of 512×512 with the Adam optimizer [28]. For different base image matting methods, we used different batch sizes and learning rates. The batch size is set to 40 for both DIM+TAM and GCA+TAM, 24 for Index+TAM. The learning rate is set to 10^{-5} for DIM+TAM, 10^{-4} for Index+TAM and 4×10^{-4} with "poly" decay strategy for GCA+TAM where the decay rate is set to 0.9.

In the main stage, our network takes five consecutive frames $\{\mathbf{I}^{t-2}, ..., \mathbf{I}^{t+2}\}$ along with their corresponding trimaps as inputs. The motivation comes from the fact that the temporal coherence term requires alpha mattes of three consecutive frames, and each frame needs the features of its two neighboring frames for alpha matte prediction. The predicted $\{\alpha^{t-1}, \alpha^t, \alpha^{t+1}\}$ are used to compute the loss function in Eq. 6. We use Adam as the optimizer to train the network on VideoMatting108 for 30 epochs with the same input resolution 512×512. Our data augmentation strategies include random shape augmentation, such as cropping, flipping and scaling, and random color augmentation, such as hue, saturation, gamma, and JPEG compression [64]. We use "poly" decay strategy with the base learning rate of 10^{-4}, 10^{-5}, 10^{-4} and decay rate of 0.9 during main training for GCA+TAM, DIM+TAM and Index+TAM. The batch size is set to 24, 16 and 24 respectively.

3.4 Video Trimap Generation

We leverage the SOTA VOS network STM [42] to segment each frame into FR, BR, and UR. That is, we let STM track FR and UR as two different objects in a video and classify the remaining pixels that do not belong to FR and UR as BR. To obtain the ground-truth labels, we label the translucent pixels obtained in the construction of VideoMatting108 without any dilation as UR, and the pixels with alpha value equals one as FR. Additionally, we give UR a higher weight (4.0 in training) to achieve class-balanced cross-entropy loss since UR generally has fewer pixels than FR and BR.

Training parameters. Same with STM, we also utilize the two-stage training strategy. First, the network is initialized from the weight pre-trained on ImageNet [16]. We then use the DIM dataset [61] augmented with random affine transformations (rotation, translation, scaling, and shearing) to pre-train the network. The network is pre-trained for 25 epochs. We then proceed to the main training stage. The only difference is that we use a larger maximum frame skip, which is 75 frames in our implementation since the videos in VideoMatting108 are much longer compared to the VOS dataset like DAVIS [44]. We train the network for 150 epochs, where every epoch consists of 850 iterations. The maximum frame skipping is gradually increased by one every two epochs. We also utilize the "stair" learning rate strategy with the base learning rate of 10^{-5}, 5×10^{-6}, 10^{-6} and 5×10^{-7} at 40, 80 and 120 epochs, respectively. We use the batch size of 4, input resolution of 512×512, and Adam optimizer for all of our experiments.

Inference strategy. When generating trimaps for a new video that is not present in the training and validation sets, we found that online fine-tuning with the user-annotated trimaps at keyframes drastically improves the performance of the network in our case. During online fine-tuning, we treat the user-annotated trimap as the ground truth and use the same random affine transform technique to generate "fake" video sequences. Subsequently, we fine-tune the network on these sequences for 500-800 iterations with a constant learning rate of 10^{-6}. When there is more than one frame of user-defined trimaps, a bidirectional inference strategy is used to ensemble the prediction results. Please refer to our supplementary material for more details.

3.5 A New Video Matting Dataset

Lacking training data is a massive barrier to deep learning-based video matting methods. For instance, the most commonly used video matting dataset from videomatting.com [18] has only ten test clips and three clips with ground-truth mattes, which is not enough for network training. To this end, we propose our video matting dataset, **VideoMatting108**. We rely on green screen video footages to extract ground-truth alpha mattes. First, we collect 68 high-quality (1080p and 4K) footages from the Internet [50]. While these footages have diverse objects, we found that they generally lack several types of objects, such as fur, hair, and semi-transparent objects. Thus, we capture 40 green screen footages for these types of objects ourselves as the supplement. Next, we carefully extract the foreground object's alpha matte and color from the green screen footages using After Effects and BorisFX Primatte Studio [6].

Table 1: Result on VideoMatting108 validation set. GCA [35], Index [38] and DIM [61] are used as the base image matting network structures to verify the effectiveness of the TAM. The best result is in bold, the second best is underlined. "+F" indicates the single image matting method is fine-tuned on our training dataset. "+TAM" denotes we add TAM for video matting. "+TAM$_{share}$" and "+TAM$_{sep}$" denote we share / separate all convolutions in TAM, respectively. "MSDdt" denotes "MESSDdt".

Method	Loss	Narrow					Medium					Wide				
		SSDA	dtSSD	MSDdt	MSE	SAD	SSDA	dtSSD	MSDdt	MSE	SAD	SSDA	dtSSD	MSDdt	MSE	SAD
GCA+F	L_{im}	49.99	27.91	1.80	8.32	46.86	55.82	31.64	2.15	8.20	40.85	60.69	34.83	2.50	8.41	38.59
+TAM	L_{im}	46.86	26.21	1.48	7.68	44.82	54.01	29.49	1.78	7.90	39.51	59.09	32.55	2.07	8.18	37.41
+TAM$_{share}$	L_{im}	49.71	27.49	1.68	8.34	46.45	57.20	29.90	1.91	8.88	41.15	62.90	33.13	2.22	9.35	39.31
+TAM$_{sep}$	L_{im}	54.06	27.69	1.78	10.37	48.03	59.13	30.75	2.00	9.84	41.56	64.89	33.90	2.30	10.37	39.78
+TAM	L_{im}+L_{tc}	48.35	25.04	1.43	8.00	45.47	52.83	27.81	1.60	7.55	38.84	57.51	30.34	1.84	7.73	36.57
+TAM	L_{im}+L_{af}	46.87	25.70	1.47	7.70	45.22	53.00	28.97	1.72	7.73	39.47	58.08	31.97	2.00	8.05	37.47
+TAM	L_{im}+L_{tc}+L_{af}	**45.39**	**24.37**	**1.28**	**7.30**	**44.01**	**50.41**	**27.28**	**1.48**	**7.07**	**37.65**	**54.35**	**29.60**	**1.69**	**6.98**	**34.81**
Index+F	L_{im}	52.75	29.49	1.97	9.78	50.90	58.53	33.03	2.33	9.37	43.53	64.49	36.39	2.73	9.73	41.22
+TAM	L_{im}+L_{tc}+L_{af}	51.18	26.31	1.52	8.87	50.02	57.91	29.36	1.81	8.78	43.17	63.56	32.09	2.10	9.21	40.97
DIM+F	L_{im}	56.40	31.77	2.56	10.46	51.76	61.85	34.55	2.82	9.99	44.38	67.15	37.64	3.21	10.25	41.88
+TAM	L_{im}+L_{tc}+L_{af}	53.61	27.77	1.90	9.48	50.12	58.94	29.89	2.06	9.02	43.28	63.27	32.15	2.31	8.88	40.45

Table 2: Comparison between GCA+TAM and GCA [35] on the 10 test clips from videomatting.com [18] with different trimaps. The best result is in bold. Please see our supplementary material for quantitative results of each test video clip.

Method	Trimap	SSDA	dtSSD	MESSDdt
GCA+F	Narrow	39.40	30.83	1.43
GCA+TAM		**36.95**	**26.37**	**1.12**
GCA+F	Medium	44.74	33.42	1.74
GCA+TAM		**42.17**	**28.81**	**1.35**
GCA+F	Wide	50.45	36.71	2.14
GCA+TAM		**49.23**	**32.94**	**1.76**

In total, our dataset consists of 108 video clips, all in 1080p resolution. The average length of the video clip is 821 frames, significantly longer than other datasets. The foreground objects cover a wide range of categories, such as human, fluffy toys, cloth (net, lace, and chiffon), smoke and plants. The background footage usually has 3D camera motion or complex scenery, adding more challenge to our dataset. We split the dataset with 80 clips in the training set and 28 clips in the validation set. Trimaps are generated and dilated on the fly with random sized kernel from 1×1 to 51×51 during training. In the validation set, trimaps are generated by dilating transparent pixels with three different kernel sizes: 11×11 for narrow trimaps, 25×25 for medium trimaps, and 41×41 for wide trimaps.

4 EXPERIMENTAL RESULTS

In this section, we present the evaluation of our approach on the VideoMatting108 dataset. We also evaluate our trimap generation algorithm. Our computing platform for video matting related experiments was a 4 V100 GPU server with 2 Intel 6148 CPUs. Trimap generation related experiments were conducted on a 4 1080Ti GPU server with 2 E5-2678v3 CPUs.

Evaluation metrics. We employ SSDA (average sum of squared difference) and two temporal coherence metrics, namely dtSSD (mean squared difference of direct temporal gradients) and MESSDdt (mean squared difference between the warped temporal gradient) from [18] to evaluate the accuracy of the predicted video alpha mattes and their temporal coherence. Besides, we also report "MSE" (mean squared error) and "SAD" (sum of absolute difference) to verify the pixel-wise accuracy of alpha values at each frame. Lower evaluation metrics correspond to better video matting results.

Table 3: Ablation study on the temporal window size W and the number of aggregated frames used for GCA+TAM. "nF" denotes n-frames are aggregated in TAM. 2F: only uses $\mathbf{I}^{t-1}$; 3F: uses $\mathbf{I}^{t-1}$ and $\mathbf{I}^{t+1}$; 5F: uses $\mathbf{I}^{t-2}$, $\mathbf{I}^{t-1}$, $\mathbf{I}^{t+1}$ and $\mathbf{I}^{t+2}$.

W	F	SSDA	dtSSD	MESSDdt
$W = 5$	3F	49.75	24.08	1.30
$W = 7$	3F	47.59	23.53	1.19
$W = 9$	3F	52.35	24.29	1.38
$W = 7$	2F	48.51	23.81	1.30
$W = 7$	5F	51.25	24.79	1.32

Quantitative comparisons. Table 1 shows the quantitative comparisons between our video matting networks and single image matting networks on the Videomatting108 validation set. For fair comparisons, we fine-tune single image matting networks on VideoMatting108 using each video frame as the image matting training data. The learning rate and input resolution are kept the same as we train our video matting network. The results are averaged over all 28 test video clips. It can be seen that our method (denoted "+TAM" with "$L_{im}+L_{tc}+L_{af}$" in the table) consistently outperform the baseline image matting networks (denoted "GCA+F", "Index+F" and "DIM+F") on all metrics. More comparison results with other methods, such as KNN Video Matting [34] and DVM [52], can be found in our supplementary material.

Furthermore, in Table 2, the GCA+TAM network also outperforms GCA on the test dataset from videomatting.com [18]. This verifies the ability of the proposed TAM that is designed to aggregate temporal information for better video object matting results. Since GCA+TAM achieves much lower metric numbers comparing to DIM+TAM and Index+TAM, we use GCA+TAM as the default for evaluating video matting, if not mentioned otherwise.

Ablation studies. We first investigate the influence of the weight sharing in TAM (see the second to the fourth row in Table 1). Different from the widely utilized "KQV" structure without any weight sharing, in our case we empirically found out that sharing "query" and "value" weights achieves the best result compared with other weight sharing configurations. Note that the conventional "KQV" structure without weight sharing performs worse than the baseline method without TAM on the validation set. We speculate that separating all weights causes over-fitting during the training, since it does achieve lower training loss compared with our design.

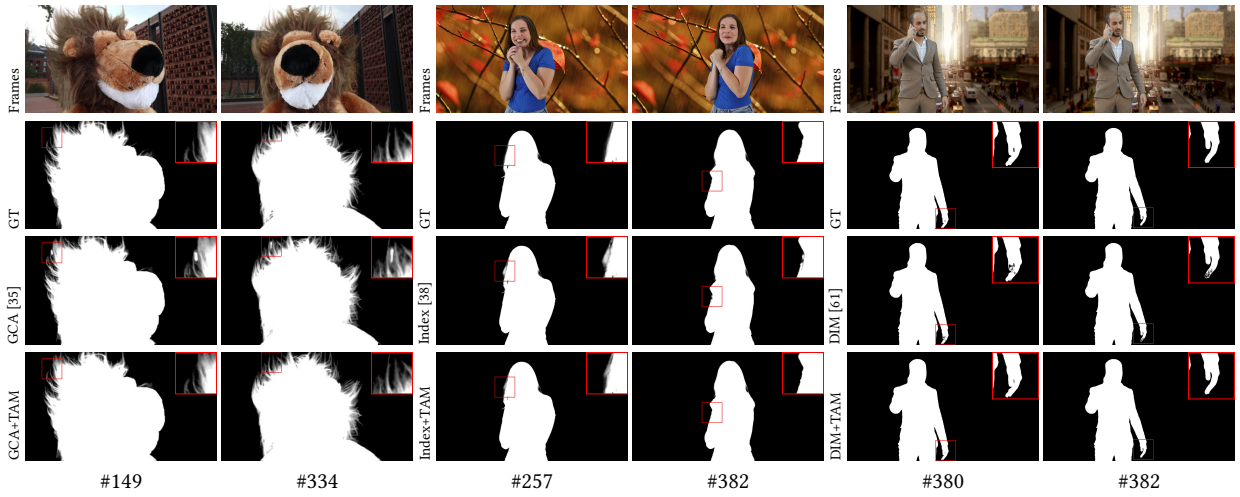

Figure 4: Qualitative evaluations that illustrate the effectiveness of our TAM. Blowups are used to show the details of the alpha matte. These three video clips are from VideoMatting108 validation set, and we use the "medium" ground-truth trimaps to obtain the results. Please see the supplementary video for the complete results.

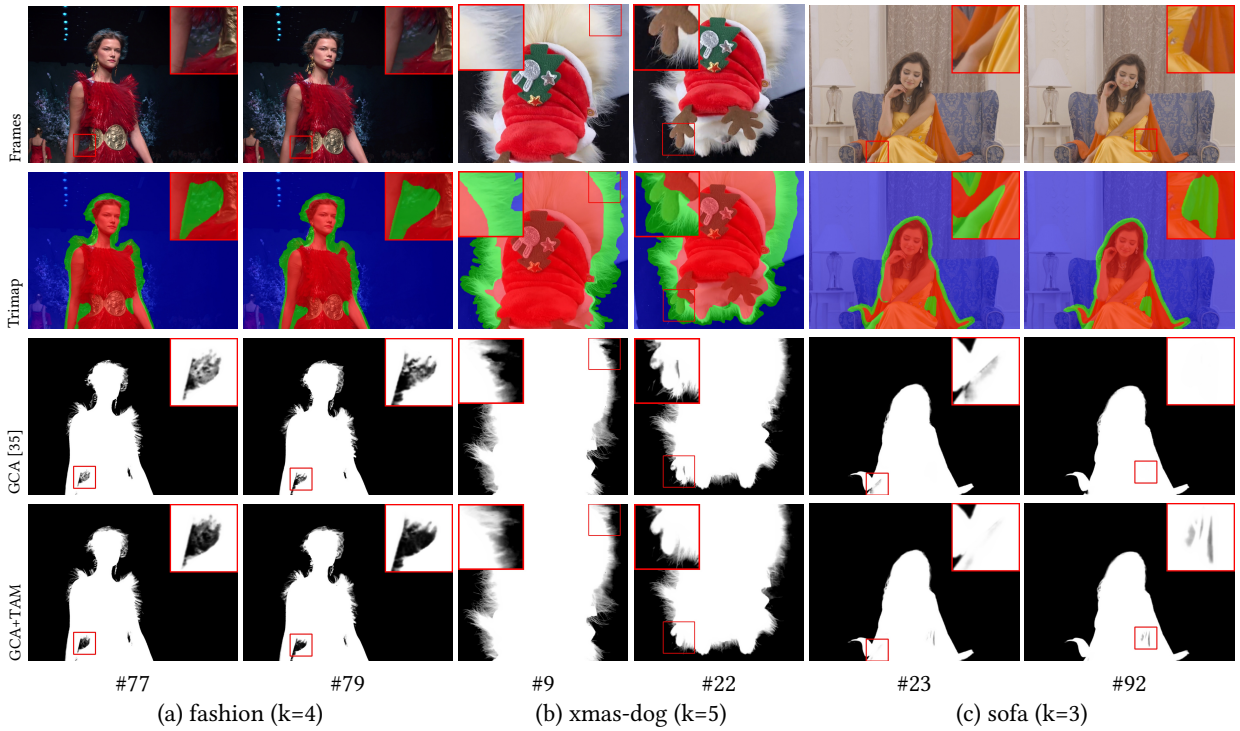

Figure 5: Comparing our GCA+TAM with GCA [35] on Internet videos. "k" indicates the number of the annotated keyframe trimaps. Please see the supplementary video for the complete results.

We proceed to assess the influence of different loss terms (see the fifth to the seventh row in Table 1). By only adding the temporal coherence term L_{tc}, we obtain performance boost across all metrics, except pixel-wise alpha value accuracy in "narrow" trimaps. Since the UR of a narrow trimap mainly consists of transparent pixels, we speculate that the motion ambiguities at these pixels are the main reason for the drop of alpha value accuracy. By only adding the target affinity term L_{af}, the temporal metrics "dtSSD" and "MESSDdt" are slightly improved while the alpha value accuracy metrics are comparable to the baseline. By combining both terms, the network achieved much improved results. For the "narrow" trimaps in particular, the direct supervision in L_{af} could suppress the erroneous affinity values, thus improving performance.

Table 4: Ablation study for trimap generation on VideoMatting108 validation set using the mIoU metric. All metrics are averaged with a per-video basis. "nFT" denotes how many keyframe trimaps are used in fine-tuning. 1FT: first frame as keyframe. 2FT: first+last frame as keyframes. 3FT: first+last+100-th frame as keyframes.

Method	FR	BR	UR	Average
STM [42]	81.43	95.58	81.63	86.21
STM+1FT	85.92	96.62	82.75	88.43
STM+2FT	87.73	97.91	84.90	90.18
STM+3FT	87.72	97.93	85.04	90.23

In Table 3, we analyze the influence of the temporal window size (Sec. 3.1) parameter W and the number of frames used in TAM. To reduce the computational cost, we conduct experiments on half of the VideoMatting108 training and validation set with "medium" ground-truth trimaps. We can see that the network performance achieves the best balance when $W = 7$. In contrast, the network performance degrades when $W = 9$. The reason may come from the difficulties of suppressing a large number of unrelated features from adjacent frames through attention. Thus, we choose $W = 7$ in all of our experiments. To verify the bi-directional design, we conduct experiments on the number of aggregated frames (fourth and fifth rows). As seen in Table 3, our bi-directional design (second row, 3F) outperforms all other configurations.

STM-based trimap generation. We use the medium trimaps from VideoMatting108 as the ground truth to evaluate the performance of STM [42] on trimap generation using the mIoU metric. As the video sequences in VideoMatting108 are long, we only use the first 200 frames in this experiment. Specifically, we choose the first, the last, and the 100-th frame as keyframes during online fine-tuning. We also gradually add their corresponding ground-truth trimaps to show their impact on online fine-tuning. The input resolution is set to 768×768, and we fine-tune the network for 500 iterations. The average time for fine-tuning STM is 7.5 minutes on one GPU. As shown in Table 4, online fine-tuning with the first or other frames improved the result by a large margin, especially in FR. Adding more ground-truth trimaps improves the results further. In conclusion, fine-tuning is necessary to adapt the network to user-specified keyframe trimaps, which improve the quality of generated trimaps. Table 4 also shows that adding more keyframes improves the mIoU score marginally, which indicates that a sparse set of annotated keyframe trimaps is enough for video trimap generation. We use around three frames in all our experiments. On the other hand, inaccurate FR/BR heavily affects the matting result. As shown in the third row of Figure 6(a), the pixels inside the gap between the two legs have erroneous high alpha values, despite the UR being roughly the same in the trimaps. In (b) and (c), the excessive URs lead to artifacts in the final alpha matte.

Qualitative evaluations. In Figure 1 and Figure 4, we show that our method could improve the temporal coherence comparing to single image matting networks. In the "plant" clip, GCA [35] produces flickering in the alpha matte although the foreground object is nearly static. As seen in the "lion" clip, GCA produces an erroneous white blob between the fur in the blowup. In contrast, our method could mitigate this discontinuity by aggregating temporal features. The same effect can be seen in the "dancing woman" clip and "standing man" clip when using Index [38] and DIM [61] as the base network, respectively. All examples validate the effectiveness of our TAM. In Figure 5, we qualitatively compare GCA+TAM with GCA on three Internet video clips. The trimaps are generated by the fine-tuned STM [42]. In the "fashion" clip, our method can alleviate the artifacts in the gap between the model's arm and body. In the "xmas-dog" and the "sofa" clip, not only our method can produce more detailed result (#9 of "xmas-dog" and #92 of "sofa"), it is also more robust to inaccurate URs in the trimap (#22 of "xmas-dog" and #23 of "sofa"). The lengths of these three clips are 89, 100 and 93 frames respectively. More results can be found in the supplementary materials.

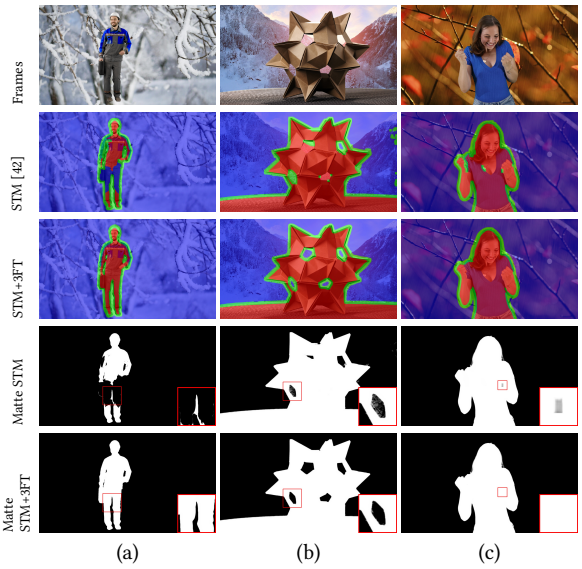

Figure 6: Qualitative evaluation of trimap generation. Red, blue and green corresponds to FR, BR and UR respectively. "3FT" denotes that we use three keyframes during fine-tuning.

5 CONCLUSION

We have developed a deep video object matting method to achieve temporally coherent video matting results. Its key feature is an attention-based temporal aggregation module to compute the temporal affinity values in feature space, which are robust to appearance changes, fast motions, and occlusions. The temporal aggregation module can be easily integrated into image matting networks to enhance video object matting performance. We constructed a video matting dataset to enable the training of video object matting and trimap generation networks. This dataset has 80 training and 28 validation foreground video sequences with ground truth alpha mattes. In the future, we plan to investigate weakly supervised video object matting methods to reduce the workload of creating high-quality video matting training data.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their constructive comments. Weiwei Xu is partially supported by National Key R&D Program of China (2017YFB1002600) and NSFC (No. 61732016). Qixing Huang would like to acknowledge the support from NSFIIS-2047677 and NSF HDR TRIPODS-1934932.

REFERENCES

[1] Yagiz Aksoy, Tunç Ozan Aydin, and Marc Pollefeys. 2017. Designing Effective Inter-Pixel Information Flow for Natural Image Matting.. In *IEEE Conf. Comput. Vis. Pattern Recog.* 228–236.

[2] Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. 2018. Semantic soft segmentation. *ACM Trans. Graph.* 37, 4 (2018), 72.

[3] N. Apostoloff and A. Fitzgibbon. 2004. Bayesian video matting using learnt image priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Vol. 1.

[4] Xue Bai and Guillermo Sapiro. 2009. A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. *IJCV* 82, 2 (2009), 113–132.

[5] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. 2015. Blind Video Temporal Consistency. *ACM Trans. Graph.* 34, 6, Article 196 (Oct. 2015), 9 pages.

[6] BorisFX. 2020. BorisFX Primatte Studio. https://borisfx.com/products/continuum-filters/primatte-studio. [Online].

[7] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jiangyu Liu, Jiaming Liu, Jiaying Liu, Jue Wang, and Jian Sun. 2019. Disentangled Image Matting. In *Int. Conf. Comput. Vis.*

[8] Youngha Chang, Suguru Saito, and Masayuki Nakajima. 2007. Example-based color transformation of image and video using basic color categories. *IEEE Transactions on Image Processing* 16, 2 (2007), 329–336.

[9] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *Int. Conf. Comput. Vis.*

[10] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. 2018. TOM-Net: Learning Transparent Object Matting from a Single Image. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[11] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. 2013. KNN matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 9 (2013), 2175–2188.

[12] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. 2016. Natural image matting using deep convolutional neural networks. In *Eur. Conf. Comput. Vis.* Springer, 626–643.

[13] Inchang Choi, Minhaeng Lee, and Yu-Wing Tai. 2012. Video Matting Using Multi-frame Nonlocal Matting Laplacian. In *Computer Vision – ECCV 2012*. Springer, 540–553.

[14] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. 2002. Video Matting of Complex Scenes. *ACM Trans. Graph.* 21, 3 (July 2002), 243–248.

[15] Yung-Yu Chuang, B Curless, DH Salesin, and R Szeliski. 2001. A Bayesian approach to digital matting. In *CVPR, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 2. IEEE, II–II.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 248–255.

[17] Martin Eisemann, Julia Wolf, and Marcus A Magnor. 2009. Spectral video matting.. In *VMV*. 121–126.

[18] Mikhail Erofeev, Yury Gitman, Dmitriy Vatolin, Alexey Fedorov, and Jue Wang. 2015. Perceptually Motivated Benchmark for Video Matting. In *BMVC*. BMVA Press, Article 99, 12 pages. https://doi.org/10.5244/C.29.99

[19] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. 2016. *A Cluster Sampling Method for Image Matting via Sparse Coding*. Springer International Publishing. 204–219 pages.

[20] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.* 3146–3154.

[21] Eduardo SL Gastal and Manuel M Oliveira. 2010. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*. Wiley Online Library, 575–584.

[22] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. 2005. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, Vol. 2005. 423–429.

[23] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. 2011. A global sampling method for alpha matting. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 2049–2056. https://doi.org/10.1109/CVPR.2011.5995495

[24] Qiqi Hou and Feng Liu. 2019. Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation. In *Int. Conf. Comput. Vis.*

[25] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.* 7132–7141.

[26] Neel Joshi, Wojciech Matusik, and Shai Avidan. 2006. Natural Video Matting Using Camera Arrays. *ACM Trans. Graph.* 25, 3 (July 2006), 779–786.

[27] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson W. H. Lau. 2020. Is a Green Screen Really Necessary for Real-Time Portrait Matting? arXiv:2011.11961 [cs.CV]

[28] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Int. Conf. Learn. Represent.* http://arxiv.org/abs/1412.6980

[29] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *Eur. Conf. Comput. Vis.*

[30] Manuel Lang, Oliver Wang, Tunc Aydin, Aljoscha Smolic, and Markus Gross. 2012. Practical Temporal Consistency for Image-based Graphics Applications. *ACM Trans. Graph.* 31, 4, Article 34 (July 2012), 8 pages.

[31] Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. 2010. Temporally coherent video matting. *Graphical Models* 72, 3 (2010), 25 – 33.

[32] Anat Levin, Dani Lischinski, and Yair Weiss. 2006. A closed form solution to natural image matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Vol. 1. IEEE, 61–68.

[33] Anat Levin, Alex Rav-Acha, and Dani Lischinski. 2008. Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 10 (2008), 1699–1712.

[34] Dingzeyu Li, Qifeng Chen, and Chi-Keung Tang. 2013. Motion-Aware KNN Laplacian for Video Matting. In *Int. Conf. Comput. Vis.*

[35] Yaoyi Li and Hongtao Lu. 2020. Natural image matting via guided contextual attention. In *AAAI*, Vol. 34. 11450–11457.

[36] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L. Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2021. Real-Time High-Resolution Background Matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8762–8771.

[37] Jinlin Liu, Yuan Yao, Wendi Hou, Miaomiao Cui, Xuansong Xie, Changshui Zhang, and Xian-Sheng Hua. 2020. Boosting semantic human matting with coarse annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8563–8572.

[38] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. 2019. Indices Matter: Learning to Index for Deep Image Matting. In *Int. Conf. Comput. Vis.*

[39] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. 2018. AlphaGAN: Generative adversarial networks for natural image matting. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 259. http://bmvc2018.org/contents/papers/0915.pdf

[40] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F. Hughes, and Frédo Durand. 2005. Defocus Video Matting. *ACM Trans. Graph.* 24, 3 (July 2005), 567–576.

[41] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. 2019. Fast user-guided video object segmentation by interaction-and-propagation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5247–5256.

[42] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. 2019. Video object segmentation using space-time memory networks. In *Int. Conf. Comput. Vis.* 9226–9235.

[43] Sylvain Paris. 2008. Edge-Preserving Smoothing and Mean-Shift Segmentation of Video Streams. In *Eur. Conf. Comput. Vis.* Springer Berlin Heidelberg, Berlin, Heidelberg, 460–473.

[44] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[45] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. 2020. Attention-Guided Hierarchical Structure Aggregation for Image Matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[46] Mark A Ruzon and Carlo Tomasi. 2000. Alpha estimation in natural images. In *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 1018.

[47] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2020. Background Matting: The World is Your Green Screen. In *IEEE Conf. Comput. Vis. Pattern Recog.* 2291–2300.

[48] Heung-Yeung Shum, Jian Sun, Shuntaro Yamazaki, Yin Li, and Chi-Keung Tang. 2004. Pop-up Light Field: An Interactive Image-based Modeling and Rendering System. *ACM Trans. Graph.* 23, 2 (2004), 143–162.

[49] Mikhail Sindeev, Anton Konushin, and Carsten Rother. 2013. Alpha-Flow for Video Matting. In *ACCV*. 438–452.

[50] Storyblocks. 2020. Storyblocks. https://www.storyblocks.com/. [Online].

[51] Jian Sun, Jiaya Jia, Chi Keung Tang, and Heung Yeung Shum. 2004. Poisson matting. *ACM Trans. Graph.* 23, 3 (2004), 315–321.

[52] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. 2021. Deep Video Matting via Spatio-Temporal Alignment and Aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6975–6984.

[53] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.* 2818–2826.

[54] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. 2019. Learning-Based Sampling for Natural Image Matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[55] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12347)*. Springer, 402–419. https://doi.org/10.1007/978-3-030-58536-5_24

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Adv. Neural Inform. Process. Syst.* 5998–6008.

[57] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. 2019. FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[58] Jue Wang, Michael F Cohen, et al. 2008. Image and video matting: a survey. *Foundations and Trends® in Computer Graphics and Vision* 3, 2 (2008), 97–175.

[59] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.* 7794–7803.

[60] Jiangjian Xiao and Mubarak Shah. 2005. Accurate motion layer segmentation and matting. In *CVPR*, Vol. 2. 698–703.

[61] Ning Xu, Brian L Price, Scott Cohen, and Thomas S Huang. 2017. Deep Image Matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Vol. 2. 4.

[62] Zongxin Yang, Yunchao Wei, and Yi Yang. 2020. Collaborative video object segmentation by foreground-background integration. In *Eur. Conf. Comput. Vis.*

[63] Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. 2020. Representative Graph Neural Network. In *Eur. Conf. Comput. Vis.*

[64] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. 2020. Context Prior for Scene Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.* 12416–12425.

[65] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Eur. Conf. Comput. Vis.* 325–341.

[66] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.* 1857–1866.

[67] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. 2020. Mask Guided Matting via Progressive Refinement Network. arXiv:2012.06722 [cs.CV]

[68] Yuhui Yuan, Xilin Chen, and Jingdong Wang. 2020. Object-Contextual Representations for Semantic Segmentation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 12351)*. Springer, 173–190. https://doi.org/10.1007/978-3-030-58539-6_11

[69] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. 2018. Context encoding for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.* 7151–7160.

[70] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. 2019. A Late Fusion CNN for Digital Matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*

[71] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. 2018. Psanet: Point-wise spatial attention network for scene parsing. In *Eur. Conf. Comput. Vis.* 267–283.

[72] Fenfen Zhou, Yingjie Tian, and Zhiquan Qi. 2020. Attention Transfer Network for Nature Image Matting. *IEEE Transactions on Circuits and Systems for Video Technology* (2020).

[73] Jieie Zhu, Miao Liao, Ruigang Yang, and Zhigeng Pan. 2009. Joint depth and alpha matte optimization via fusion of stereo and time-of-flight sensor. In *CVPR*. 453–460.

[74] Dongqing Zou, Xiaowu Chen, Guangying Cao, and Xiaogang Wang. 2020. Unsupervised Video Matting via Sparse and Low-Rank Representation. *PAMI* 42, 6 (2020), 1501–1514.