This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI 10.1109/TMM.2022.3150177, IEEE Transactions on Multimedia

A Coarse-to-Fine Framework for Automatic Video Unscreen

Anyi Rao, Linning Xu, Zhizhong Li, Qingqiu Huang, Zhanghui Kuang, Wayne Zhang, and Dahua Lin

***Abstract*—Video unscreen, a technique to extract foreground from given videos, has been playing an important role in today's video production pipeline. Existing systems developed for this purpose which mainly rely on video segmentation or video matting, either suffer from quality deficiencies or require tedious manual annotations. In this work, we aim to develop a fully automatic video unscreen framework that is able to obtain high-quality foreground extraction without the need of human intervention in a controlled environment. Our framework adopts a coarse-to-fine strategy, where the obtained background estimate given an initial mask prediction in turn helps the refinement of the mask by the alpha composition equation. We conducted experiments on two datasets, 1) the Adobe's Synthetic-Composite dataset, and 2) DramaStudio, our newly collected large-scale green screen video matting dataset, exhibiting the controlled environments. The results show that the proposed framework outperforms existing algorithms and commercial software, both quantitatively and qualitatively. We also demonstrate its utility in person replacement in videos, which can further support a variety of video editing applications.**

***Index Terms*—Automatic video unscreen; amateur green screen matting, background estimation.**

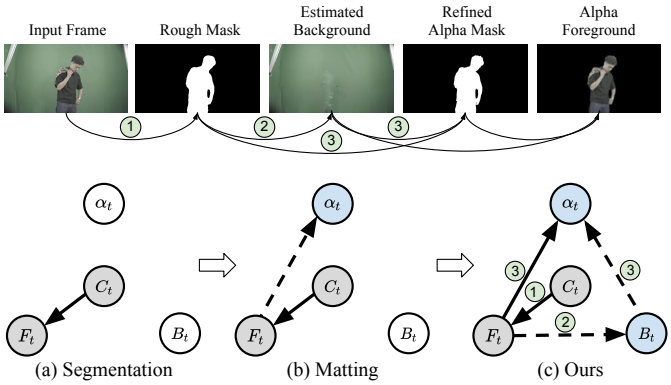

Fig. 1: Automatic video unscreen procedure comparison among (a) segmentation, (b) matting, and (c) our unified framework. Each node represents one variable in the composition equation. The unused nodes are not colored. The dashed lines indicate the newly established information flow when compared from left to right. Each video frame C_t is softly blended by the foreground F_t and the background B_t, controlled by the alpha-mask α_t. A brief procedure of the proposed coarse-to-fine framework is illustrated with green circles.

I. Introduction

Video unscreen has been an indispensable part of modern video production pipelines. Recording a video on the spot is often a highly costly activity. Film producers thus may resort to more affordable approaches, *e.g.*, recording the video in a controlled environment and then "migrate" the foreground to the desired scene during post-production. In recent years, with the thriving of online video services, video unscreen techniques have seen demands emerging in new domains, *e.g.*, the production of user-generated content and video conferencing. As the professional video editing procedures are overly complicated and cumbersome for an ordinary user, and lengthy human intervention is simply infeasible in the online service settings, these applications lead to a significant challenge—they require the system to work in a *completely automated* manner. We note that, while there has been plenty of sophisticated image unscreen tools [1], video unscreen remains relatively new [2], [3], and is rarely possible to be fully automatic. In this work, we focus on a controlled yet common scenario, where the unscreen target is positioned in front of a relatively clean background. To be practical and applicable to our daily life environments, the developed method should be less sensitive to nonuniform lighting, and be robust to noisy backgrounds, eliminating the requirement for a professional movie studio.

Existing video unscreen techniques primarily rely on video segmentation or video matting. However, both approaches are limited. The techniques based on foreground segmentation lack the capability of preserving fine details [4], *e.g.*, human joints and hairs. Video matting relaxes the binary partition assumption into an alpha composition model [5],

$$C_t = \alpha_t \otimes F_t + (1 - \alpha_t) \otimes B_t, \tag{1}$$

where each video frame C_t is considered as a *soft blending* of the foreground F_t and the background B_t, controlled by the alpha-mask α_t. Despite the fact that matting techniques [6]–[8] have shown their capability of producing results of substantially higher quality, their application in the context of video unscreen faces a significant difficulty, namely, they require a trimap for each frame to be provided by the user. Recent works [9], [10] attempted to mitigate this problem by requiring the background instead of the trimaps. This way still faces practical difficulties in real-world applications, where the backgrounds are hard to acquire.

In this work, we propose an automatic video unscreen framework aiming to tackle the challenges mentioned above. This framework adopts the alpha-composition model at its core, delivering high-quality results while eliminating the

Anyi Rao, Linning Xu, Qingqiu Huang and Dahua Lin are with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China (e-mail: {ayrao, linningxu, hq016, dhlin}@ie.cuhk.edu.hk).

Zhizhong Li, Zhanghui Kuang and Wayne Zhang are with SenseTime Research, Hong Kong, China (email: {lizz, kuangzhanghui, wayne.zhang}@sensetime.com).

need for human intervention—users are no longer required to provide neither the trimaps nor the background. The key idea is to untie the coupling between the background and alpha matte, turning it into a coarse-to-fine refinement process. As illustrated in Figure 1, the initial prediction of the alpha mask is then used in estimating the background, which in turn helps further refinement of the mask. In this way, the framework achieves both desiderata at the same time, namely *high-quality output* and *free of user intervention*.

To test the effectiveness of the proposed techniques in comparison with others, we construct *DramaStudio*, a large-scale dataset comprised of real-world video recorded from drama studios with foreground masks annotated by professionals. This dataset contains 420 videos with over $334K$ annotated frames, which covers around several hundred annotated people. On both DramaStudio and Adobe's Synthetic-Composite dataset, the proposed framework outperforms mainstream methods under the *background-free* and *trimap-free* setting. Our empirical studies also show that the refinement process can effectively cope with the flaws in the trimap and background estimates, endowing the system with the robustness needed in real-world applications.

Overall, our contributions lie in two aspects: 1) We develop an automatic video unscreen framework that can produce high-quality foreground extraction without the provision of trimaps and backgrounds. This is accomplished by the coarse-to-fine refinement process built on top of the alpha composition model. 2) We construct DramaStudio, which provides a large-scale collection of frames with high-quality mask annotations.

II. Related Work

Video Segmentation. Segmentation can be viewed as a rough solution to automatic video unscreen. Per-frame image-based segmentation [11]–[19] is not ideal for videos as temporal constraints can be violated. Video segmentation improves accuracy by exploiting the temporal relations in the video sequence with propagation and sampling. They can be categorized into two branches: unsupervised and semi-supervised methods. In the semi-supervised setting, ground-truth masks are assumed to be given in the first frame. Some works [20]–[23] propagate flow, mask or semantic labels to unlabeled frames. Others like FeelVos [24] and STM [25] use a matching mechanism or memory networks to fuse information of multiple frames to improve the segmentation accuracy. Unsupervised video segmentation cannot rely on any supervision at inference time. Many approaches take advantage of the motion patterns of objects as complementary cues [26]–[29], which take a two-stream network to process the RGB image and the corresponding optical flow separately and fuse the results in the end. To avoid the expensive computation of optical flow, some works [30]–[33] utilize higher-order spatial and temporal relations between video frames to bring more comprehensive content understanding. Other approaches [34], [35] directly feed consecutive frames into the networks to learn rich semantic relations through cross-frame correlations. However, these video-based segmentation methods are prone to accumulate errors calling for a new system with high-accuracy performance and robustness on each frame.

As the performance of segmentation model backbones becomes saturated, improvements on standard benchmarks such as DAVIS [36], Cityscapes [37] and YouTube-VOS [38] are stagnated. The research focus is shifting to improving the efficiency of video segmentation [39]. Xu *et al* [40] use an adaptive keyframe selection policy, and Jain *et al* [41] fuse predictions of the keyframe from a large model and other frames from a compact model. Liu *et al* [4] observe that keyframe based methods might produce different qualities for keyframes and other frames, so they impose the temporal consistency constraint during training and apply a per-frame prediction scheme in inference. In this work, segmentation serves as the initialization of a rough foreground. We show that even lightweight networks in this step can yield high-quality foreground at the end, making the video unscreen system practically applicable.

Video Matting. With the help of trimaps, video matting predicts a detailed alpha matte which can be used to recover the mixing factor of foreground and background [42]. Similar to video segmentation, traditional video matting methods [43], [44] impose temporal coherency by propagation and sampling. There is barely any new video matting in the deep learning era, as is embodied by the fact that methods tested on the benchmark [45] are basically image-based. User-provided trimaps are important for both conventional, non-learning based methods [46]–[48], and learning-based methods [6], [49], [50]. To further improve the alpha matte prediction, IM [7] designs an index-guided upsampling, CAM [51] predicts both the alpha matte and the foreground, and FBAM [8] predicts the alpha matte, foreground, and background simultaneously. They achieve SOTA performances with high-quality trimap inputs, but they are not robust to faulty user-generated trimaps [9].

To make the system automatic, researchers consider using video segmentation to generate rough trimaps. We note that these works are mainly human-focused, ranging from portrait matting [52], [53] to whole body matting [9], [54]–[57]. However, they tend to fail in general daily-life scenarios when people are interacting with objects, *e.g.*, people wearing accessories, sloppy outfits, or holding papers. Our proposed system is able to handle these complicated settings. What's more, it can also output the background as a byproduct to facilitate further applications such as person replacement.

Video Completion. Video inpainting aims to fill the missing regions in a video sequence with both spatial and temporal consistency. It recovers the background video given the foreground mask from each frame. Traditional methods usually complete regions by patch matching in 3D [58] and 2D [59], [60]. Deep neural networks combine 3D and 2D convolutions to learn how to collect information from the reference frames to generate the missing contents. Several works use 3D/2D CNN [61], [62] or transformer [63] for feature extraction and content reconstruction but are extremely memory-consuming. Flow-based methods [64], [65] employ optical flow to guide the inpainting and fill the remaining pixels with pre-trained image completion models. Most recently, Ke *et al* [66] include occlusion awareness and Ouyang *et al* [67] apply internal learning to improve performance. Different from the aforemen-

tioned methods, our framework targets to extract foreground from its background, which faces more challenges to handle the dynamic motion of the foreground, while the background only servers as a byproduct to help the unscreen.

III. Automatic Video Unscreen

Our video unscreen system is both trimap-free and background-free (See Figure 2 for the illustrated pipeline). Recall the composition equation (1), where the observed color image of the t-th frame C_t is composed of the unknown foreground F_t, background B_t and alpha matte α_t. Our goal is to estimate the alpha foreground $\alpha_t \otimes F_t$. We iteratively estimate all these variables at each timestamp. The system starts from a rough mask predicted by segmentation (for $t = 1$) or binarized from last-frame prediction (for $t > 1$). It then refines the alpha foreground by fusing high-level semantic information from coarse prediction and low-level spatial information from background prediction. Finally, it obtains the fine-grained alpha foreground using Equation (1). In this process, modules are complementary to each other and the whole pipeline achieves a promising performance with lightweight components.

In the following, we detail three main modules: 1) **Coarse prediction.** It predicts a coarse foreground based on image segmentation and matting, containing the foreground semantics learned from massive data. 2) **Background estimation**. This part reconstructs the background by inpainting or GMM, which utilizes the temporal and spatial constraints. 3) **Final prediction with ensembled masks**. It generates a fine-grained foreground based on the coarse foreground and the reconstructed background.

A. Coarse Prediction

Initial Segmentation. Suppose a video V in resolution $w \times h$ has n frames $V = \{C_1, C_2, \cdots, C_n\}$. At the first timestamp, the initial binary mask $M_1^{\text{seg}} \in \mathbb{R}^{w \times h}$ for C_1 is obtained from a segmentation network (Deeplab v3+ [11] in our experiments). Initial masks for later frames C_t, $t > 1$ come from the binarization of the previous frame's final alpha matte prediction $\alpha_{t-1} \in \mathbb{R}^{w \times h}$. Note that the video clips we dealt with are assumed to be single-shot. Multiple-shot videos are cut into single-shot videos with shot detection [68]–[70].

Connected Area Filtering. To reduce noise in the initial segmentation mask M_t^{seg}, we use contour detection [71] to detect all connected areas $\{\mathcal{O}_t^1, \cdots, \mathcal{O}_t^i\}$ in the foreground and apply noise filtering. Each connected area is represented as a binary mask where the value is 1 inside the area and 0 outside. We utilize a saliency score $S_{t,i}$ to encourage that larger objects get higher scores and small noisy points get lower scores.

$$S_{t,i} = \frac{\sum_x^w \sum_y^h \mathcal{O}_t^i}{w \times h}, \tag{2}$$

where $\sum_x^w \sum_y^h$ sums over spatial dimensions. The mask after connected area filtering is

$$M_t^{\text{filt}} = M_t^{\text{seg}} \otimes \left(\sum_i \gamma_i \mathcal{O}_t^i \right), \tag{3}$$

where $\gamma_i = 1$ if $S_{t,i} > \lambda$, and $\gamma_i = 0$ otherwise, λ is a pre-defined hyperparameter.

Trimap Free Matting. Referring to the auto trimap generation procedure in [9], we dilate and erode the mask and take the inconsistent region between the dilated and eroded masks as unknown in the trimap. Then we apply a matting network and get a coarse mask M_t^{coar}. We adopt DIM [6], a variant of UNet [72] here. It is a relatively small network, which aligns our motivation of using a series of lightweight modules in our framework to show the effectiveness of this system design. We do not require each module to use the SOTA models on single tasks (which always rely on deeper networks or more complex modules to achieve high accuracy). By integrating lightweight and relatively good ones into our designed framework, it may achieve even better and more robust performance.

B. Background Estimation

The coarse mask M_t^{coar} coming from segmentation and matting network tends to be overly smooth. We compute another foreground mask M_t^{spat} together from the view of background reconstruction, and use them to supplement the pixel-level details. We experiment with two simple and effective background prediction methods: 1) GMM color filtering, which achieves superior performance on relatively simple and clean background situations, and 2) traditional region fill inpainting method to handle arbitrary complex background.

Color Prior Background Estimation. In common drama studio scenarios, the background is relatively clean (*e.g.*, green/white/blue mat). Given a coarse mask of foreground and background, we set up the color prior as two Gaussian Mixture Models (GMMs): one is fit for the foreground, and the other is fit for the background. The probability of each pixel X_t belonging to the foreground is approximated by

$$P\left(X_t\right) = \frac{P_{\text{fg}}(X_t)}{P_{\text{fg}}(X_t) + P_{\text{bg}}(X_t) + \epsilon}, \tag{4}$$

where $\epsilon = 10^{-6}$, P_{fg} and P_{bg} are the pdf of the foreground and background GMM respectively. The mask M_t^{spat} at pixel X_t is set to $P\left(X_t\right)$. To reconstruct B_t, we fill the missing pixels of the background image by the weighted average of Gaussian means in the background mixture model. We implement the GMM according to [73], and update its parameters adaptively at each time t. For better performance, HSV color space is used here instead of RGB.

Inpainting Based Background Estimation. When the video resolution is high (1080P) and the background is complicated, the inpainting region could be very large (sometimes over 80%). It is time-consuming to use deep learning inpainting methods [65], [74], [75]. We choose a traditional method region fill [76] f_{inp} to inpaint our background. For each frame,

$$B_t = f_{\text{inp}}\left(\frac{1}{w} \sum_{t-w}^{t} \left(1 - f_{\text{dil}}(M_t^{\text{seg}})\right) \otimes C_t \right), \tag{5}$$

where f_{dil} refers to the dilation function used in auto trimap generation, aiming to remove noise near the boundary of M_t^{coar}, w is the length of the temporal sliding window.

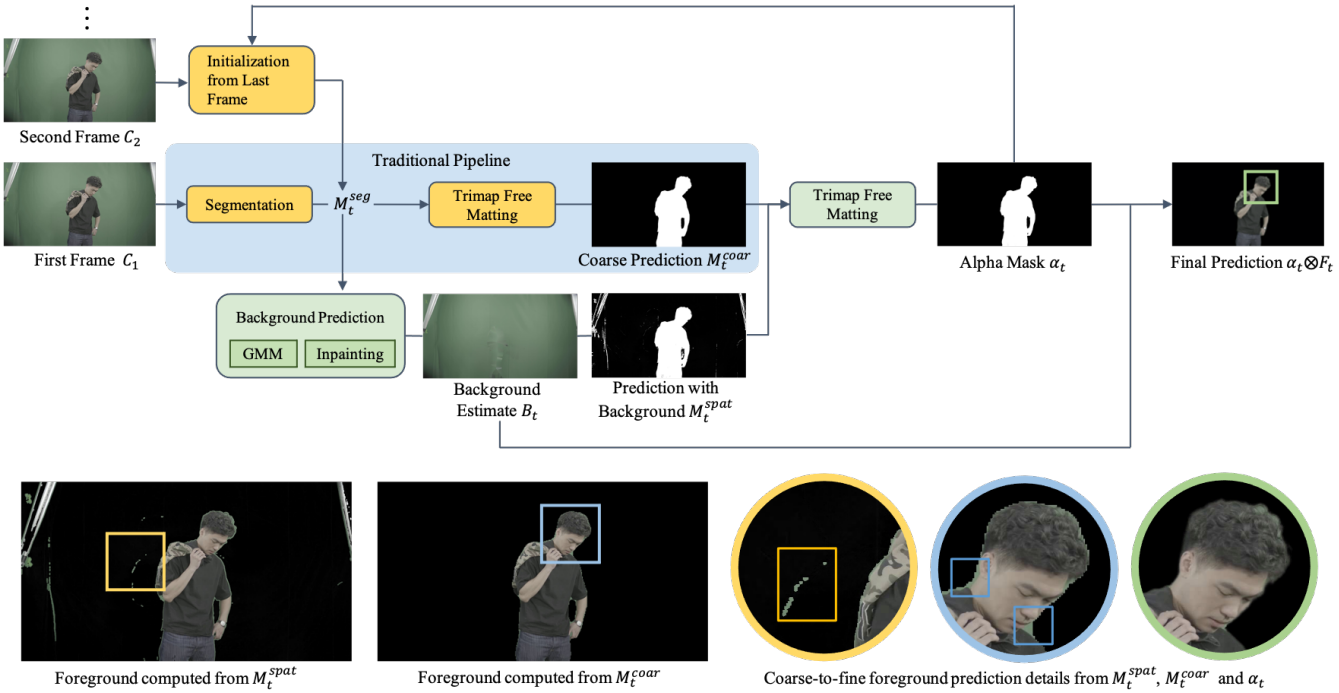

Fig. 2: The pipeline of the proposed automatic video unscreen system. The coarse prediction M_t^{coar} has semantic information but the boundary is not perfect. The prediction with background information M_t^{spat} provides fine-grained boundary information but is noisy. Integrating them M_t^{ense} produces a better result. The detailed comparison is shown at the bottom.

Specifically, in static single-shot cases, $t - w = 1$. This equation only relies on previous frames, so it is suitable for the online prediction setting. With the estimated background B_t, we compute the mask M_t^{spat} using f_{sub}, which is an adaptive background subtraction function applying on each pixel X_C and X_B on the frame C_t and estimated background B_t as

$$f_{\text{sub}}(X_C, X_B) = \min\left(1, \frac{\|X_C - X_B\|_2}{\delta \|X_C\|_2 + \epsilon}\right) \tag{6}$$

where δ is a hyper-parameter and we set it to 0.5 in the experiments, and $\epsilon = 10^{-6}$. It prevents the subtraction fails when the foreground or background is black.

C. Final Prediction with Ensembled Mask

An ensembled mask M_t^{ense} is obtained by taking the intersection as,

$$M_t^{\text{ense}} = M_t^{\text{coar}} \otimes M_t^{\text{spat}}. \tag{7}$$

Considering the intersection and the M_t^{spat} prediction may result in holes in M_t^{ense}, based on M_t^{spat}, we apply the same operation as in trimap free matting to get the final alpha matte prediction α_t. The dilation and erosion step will allow the hole areas to be labeled as uncertain areas in the trimap and filled by the matting process. Finally, the alpha foreground $\alpha_t \otimes F_t$ can be calculated from the composition equation (1) explicitly,

$$\alpha_t \otimes F_t = C_t - (1 - \alpha_t) \otimes B_t. \tag{8}$$

IV. Experiments

A. Experiments Setup

Data. We test all the baseline methods on 1) *DramaStudio* dataset and 2) Synthetic-Composite Adobe Dataset [6], [9]. The DramaStudio is for the human-centric common-life green-screen application scenario with the largest amount of annotated frames compared to existing ones, as shown in Table I. Among our 420 videos, *train* and *test* sets are divided into 381 and 39, respectively. Compared to BGM V2 dataset [10], our cases are more complicated, including the non-uniform lighting and noisy environments. Since SyntheticAdobe [6] only provides image matting annotation, we follow the evaluation protocol in BGM [9] and build a video version. Specifically, for the testing set, there are 11 held-out mattes of human subjects composed with the 9 background image provided in [9]. We continuously shift the foreground and keep the background image still, resulting in 99 testing videos with $\sim$30,000 frames in total.

Implementation details. We train these models for 60 epochs with SGD, where the batch size is 64 and the momentum is 0.9. The initial learning rate is 0.001 and the learning rate is divided by 10 every 20 epochs. We adopt Deeplab V3+ as the segmentation network and DIM as the matting network. They are trained separately since the auto trimap generation is not differentiable. The label to train segmentation networks comes from the binarization of annotated alpha matte, *i.e.*, using 128 as the threshold for alpha values ranging from 0 to 255.

Evaluation metrics. We take four commonly used metrics: SAD error, MSE error, connectivity error, and M_{iou} [9]. All the video frames are tested under their original resolution, *i.e.*, 1920×1080 in landscape mode and 1080×1920 in portrait mode. Since the user-defined trimap is unavailable in our automatic video unscreen setting, all metrics are computed over the whole image. This is similar to the evaluation of foreground composite in [8] but is different from previous

TABLE I: Dataset comparisons. *Frames* means effective annotation frames. *Duration* is in second. *Clips* is the number of video clips.

	Frames	Duration/(s)	Clips
SyntheticAdobe [6]	223	-	-
VideoMatting [45]	838	35	5
BGM V2 [10]	240,709	10,029	484
DramaStudio	**334,402**	**13,950**	**420**

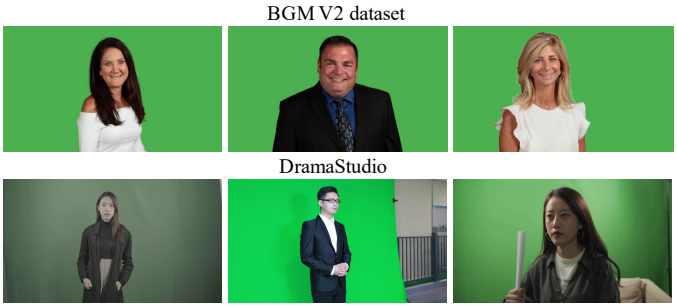

Fig. 3: Samples from BGM V2 dataset and DramaStudio. Our dataset contains samples with various lighting conditions and inconsistent environment settings.

matting evaluations that only test on the uncertain region of the given trimap. Note that M_{iou} needs to binarize the masks before computing the overlapping areas, so it is not sensitive to subtle boundary changes. In our evaluation, we mainly consider the SAD, MSE, and the connectivity error, and take the M_{iou} as a reference.

B. Comparing Methods

We adapt existing segmentation and matting methods into the auto unscreen setting, *i.e.*, given a video as the sole input, the system targets to output the corresponding alpha foreground video.[1]

Video segmentation. We test two commonly used SOTA segmentation methods, the image-based Deeplab V3+ [11] and the video-based STM [25]. The final foreground comes from the element-wise multiplication $F_t = M_t^{\text{seg}} \otimes C_t$. For a fair comparison, STM is further adapted to a video matting version STM-Mat after removing the argmax operation in the end to output a soft 0-1 foreground probability as the prediction.

Video matting. DIM [6], IM [7], AdaM [49], CAM [51] and FBAM [8] are trimap-based deep matting techniques. Among them, FBAM predicts alpha and alpha foreground composite concurrently. We modify them into a trimap-free matting method by feeding an auto-generated trimap. LFM [54], HAtt [55], BSHM [56] are trimap-free portrait matting methods. BGM [9] is a recent trimap-free SOTA matting method and requires an additional input of the background image, so we apply our background estimation to provide the background image to it. All alpha foregrounds are computed by Equation (8), in which B_t is also the same as what we use for a fair comparison. [2]

For Deeplab V3+ [11], STM [25], CAM [51], FBAM [8] and BGM [9], we use their original implementations. For DIM [6] and IM [7], we take the implementation in MMEditing[3], which achieves better performance than the original implementation. We reproduce the methods [49], [55], [56] that have no publicly available codes. The quantitative evaluation is conducted on the predicted mask: M^{seg} for segmentation, and α for matting and our method. The qualitative user study

[1]Existing commercialized software [2], [3] do not support large-scale calling.

[2]For FBAM, the alpha foreground $\alpha_t F_t$ comes from its own prediction.

[3]https://github.com/open-mmlab/mmediting

TABLE II: Overall results. The magnitude of MSE error and Conn error is 10^3 while the magnitude of M_{iou} is 10^{-2}.

(a) Results on DramaStudio.

Settings	SAD (↓)	MSE(↓)	Conn(↓)	M_{iou}(↑)
DeepLab V3+ [11]	95.67	121.75	132.26	87.75
STM [25]	151.22	201.68	211.84	83.10
STM-Mat [25]	149.66	186.16	208.85	83.12
DIM [6]	92.13	115.52	127.05	88.34
IM [7]	92.63	115.91	127.90	88.28
AdaM [49]	92.35	115.66	127.17	88.81
CAM [51]	92.20	115.91	127.65	88.76
LFM [54]	93.18	117.04	129.04	86.89
HAtt [55]	92.81	116.46	128.16	88.45
BSHM [56]	92.53	116.35	127.90	88.30
FBAM [8]	92.06	115.38	127.23	88.36
BGM [9]	140.65	182.43	198.13	81.76
BGM V2 [10]	93.09	116.48	127.18	87.48
Ours	**74.86**	**81.86**	**96.61**	**90.86**

(b) Results on SyntheticAdobe.

Settings	SAD (↓)	MSE(↓)	Conn(↓)	M_{iou}(↑)
DeepLab V3+ [11]	76.77	101.69	115.74	89.96
STM [25]	248.13	359.73	377.29	71.01
STM-Mat [25]	248.48	342.49	372.77	71.03
DIM [6]	72.44	91.82	106.26	90.76
IM [7]	72.33	94.37	109.16	90.51
AdaM [49]	71.98	95.66	127.17	88.81
CAM [51]	71.60	94.91	107.65	88.76
LFM [54]	72.92	95.04	119.04	86.89
HAtt [55]	71.81	93.46	111.16	88.45
BSHM [56]	72.53	94.35	117.90	88.30
FBAM [8]	71.25	92.38	107.70	90.61
BGM [9]	96.03	129.43	147.85	88.01
BGM V2 [10]	72.32	92.61	109.49	89.49
Ours	**69.27**	**90.01**	**104.36**	**90.92**

TABLE III: Overall user study results.

(a) Results on DramaStudio.

Ours vs.	much better	better	similar	worse	much worse
STM [25]	55%	45%	0%	0%	0%
DIM [6]	34%	42%	19%	5%	0%
IM [7]	35%	38%	20%	7%	0%
FBAM [8]	33%	40%	17%	10%	0%
BGM V2 [10]	35%	48%	16%	1%	0%

(b) Results on SyntheticAdobe.

Ours vs.	much better	better	similar	worse	much worse
STM [25]	42%	38%	20%	0%	0%
DIM [6]	27%	29%	41%	3%	0%
IM [7]	26%	32%	40%	2%	0%
FBAM [8]	24%	28%	43%	5%	0%
BGM V2 [10]	28%	41%	31%	0%	0%

Fig. 4: Overall qualitative comparison among different methods. The first four rows are two examples from DramaStudio, and the rest four rows are from SyntheticAdobe. Odd rows are predicted alpha mask and even rows are predicted alpha foregrounds. Best viewed in color.

is conducted on the predicted alpha foreground $\alpha_t \otimes F_t$. For any two methods, 10 users are asked to compare 10 pairs of videos side by side, and each pair of videos is processed by the two compared methods.

C. Overall Results

The overall quantitative results are shown in Table II, and the qualitative comparisons are shown in Figure 4. Table III shows the user study results.

DeepLab V3+ infers each frame independently. We take it as the baseline since all other methods take the first frame initialization based on it. STM, as one of the state-of-the-art video segmentation methods, performs inference as mask tracking and is very sensitive to temporal inconsistencies. It tends to fail on fast-moving objects and subsequently fails on all remaining frames. Thanks to the fine-grained supervision signal of annotated alpha mattes, STM-mat performs slightly better than STM. However, both of them are likely to accumulate errors with their memory design due to the lack of self-correction ability.

The adapted trimap-free version of DIM, IM and FBAM take the same auto-generated trimap as in BGM, where the image segmentation initialization comes from Deeplab V3+. Compared to DeepLab V3+, they improve 3% $\sim$ 5% on SAD, MSE, and Conn on DramaStudio. LFM, HAtt and BSHM show similar results since they are designed for human portraits and are not good at handling the details.

BGM is a SOTA trimap-free matting approach and requires a given background. The same background we predict is provided to BGM. The performance of BGM is inferior to DIM, IM and FBAM since an estimated background performs worse than an accurate background image.

With our jointly-solving design, we combine the semantics of segmentation prediction and the spatial details of background estimation to get a more accurate alpha foreground. From Table IIa, we can see that our pipeline achieves the best result among all with 20% improvement on DramaStudio. Similar conclusions can be made from the results on SyntheticAdobe in Table IIb, and the user study on both datasets in Table IIIa, IIIb. Qualitative results in Figure 4 also demonstrate the superiority of our system.

D. Ablation Study

Importance of background estimation. We conduct experiments on two datasets with different background estimation methods and present the quantitative results in Table IV. Note that, without background estimation, our method degrades to DIM [6]. With background information, our method improves the baseline by a large margin. On DramaStudio, color prior method gains 10% compared with DIM, which is better than

TABLE IV: Comparison of different background guidance.

Settings	DramaStudio		SyntheticAdobe	
	SAD (↓)	MSE (↓)	SAD (↓)	MSE (↓)
DIM [6]	92.13	115.52	72.44	91.82
BGM V2 + inpaint bg	94.31	116.94	72.32	92.61
BGM V2 + color bg	92.09	115.23	73.76	94.23
BGM V2 + flow bg [65]	83.02	98.65	**67.79**	**90.61**
Ours + inpaint bg	86.45	105.03	**69.27**	90.01
Ours + color bg	**74.86**	81.86	70.12	90.79
Ours + flow bg [65]	**74.82**	**81.81**	**68.01**	**90.79**

TABLE V: Comparison of using temporal cues at different module on DramaStudio. Here F means conducting per-frame inference while T means using temporal information.

Settings	SAD (↓)	MSE (↓)	Conn (↓)
DIM [6] Seg-F	92.13	115.52	127.05
STM [25] Seg-T	151.22	201.68	211.84
Ours + Seg-F + BG-F	82.30	103.20	111.40
Ours + Seg-T + BG-F	77.26	85.60	100.41
Ours + Seg-T + BG-T	**74.86**	**81.86**	**96.61**

the inpainting method. This is because the GMM modeling benefits from the statistics of known background pixels when the background is relatively clean. When the background is complicated, as in SyntheticAdobe, GMM brings minor improvements (∼4%) as the background recovery is hard. Region fill achieves better results since it tends to smooth the background holes and introduces less noise.

Alternative better background estimation modules are likely to bring better performance, yet at the cost of efficiency. When we apply a SOTA deep learning based background estimation method [65] (flow bg), the performance improves $\sim 1\%$ at the cost of $3\times$ memory and $10\times$ time consumption. It shows that with a tailored system design following the composition equation, our relatively simple background estimations can bring superior results than others.

Qualitatively, the benefits of background estimation is shown in Figure 5 and Figure 6. Looking at each frame individually, we can find out that the boundary details within the convex hull are much better, *e.g.*, the insider areas of the girl's hair and the arms (see the comparison in Figure 5). Looking at the video sequence shown in Figure 6, the background information could provide detailed boundaries that prevent the masks from exploding even in the presence of fast and large movements.

Additionally, we compare the byproduct of the proposed system, *i.e.*, the estimated background, with the background prediction from FBAM [8] in Figure 7. We can see that our estimated background, which considers the temporal consistency over frames, is visually much better than FBAM.

Effects of temporal consistency. Temporal information brings consistency over time. In our framework, temporal information is gathered in two modules, namely the initialization of the segmentation mask from last frame and the prediction of the background image.

To study the effects of different temporal cues, we ablate the modules and report the results in Table V. DIM [6] is a per-frame inference method without using background information. Video segmentation STM [25] uses the temporal information from the previous mask, but it is not robust to the previous faulty mask. At each step, we not only use the temporal consistency but also use the spatial information coming from the background estimation, which is more robust to fast-moving action/objects. The errors drop by 10% when we add the per-frame background estimation module (Seg-F + BG-F). When we reuse the last frame final prediction in the segmentation initialization, the performance improves by 5% (Seg-T + BG-F). And when we further update background information according to the previous frames, the framework reaches the best results (Seg-T + BG-T).

Effect of connected area filtering. As the unscreen system may produce noisy masks under complicated scenarios, we apply connected area filtering to mitigate its bad influence. To prove its effectiveness, we alter the thresholds in the connected

Fig. 5: The role of using background image M_t^{spat} at individual frame level. Best viewed in color with zoom-in.

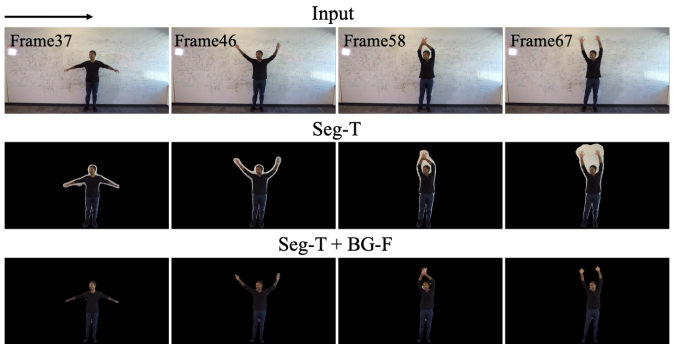

Fig. 6: The role of using background image M_t^{spat} at video sequence level. Seg-T is to use temporal information in segmentation. Seg-T + BG-F is to add per-frame background image guidance. Best viewed in color with zoom-in.

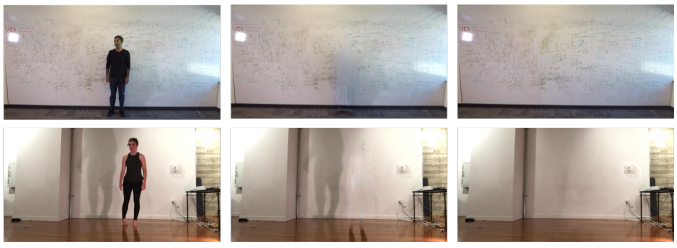

Fig. 7: Comparison of estimated backgrounds. Best viewed in color with zoom-in.

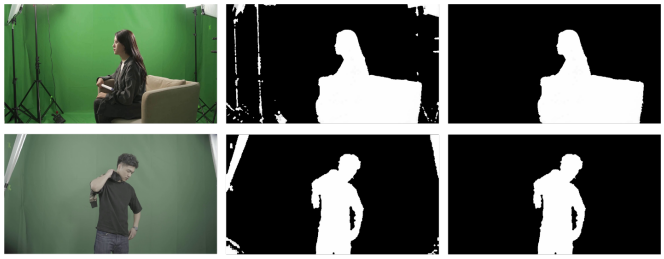

Fig. 8: Qualitative comparison on the effects of connected area filtering. Best viewed in color with zoom-in.

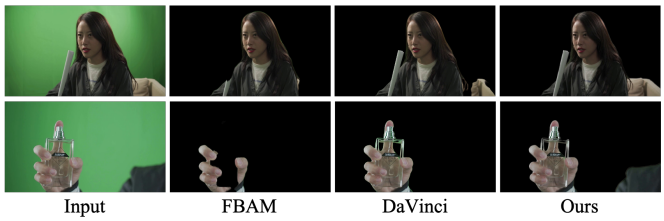

Fig. 9: Qualitative comparison with non-dl green screen software DaVinci, where we produce better results. Best viewed in color with zoom-in.

TABLE VI: Comparison of using different threshold λ for saliency score $S_{t,i}$ at connected area filtering on DramaStudio.

λ	1e-4	5e-4	1e-3	5e-3
	76.55	76.98	**74.86**	75.97

TABLE VII: Comparison on user time cost in second.

Settings	DaVinci	Ours + Interactive
Single Person (easy)	182	21
Single Person + Object (med.)	251	29
Multiple People (hard)	404	39

area filtering and show the results in Table VI. When is λ is small, the noise filtering doesn't contribute, and the SAD error is 77.91. As we increase the thresholds, the best performance 74.86 comes when λ is at a moderate number $1e{-}3$. A bigger threshold eliminates every appearing object $\mathcal{O}_t^i$, causing a performance drop. We also compare the qualitative mask before and after the connected area filtering, as shown in Figure 8. It shows that the filtering can effectively remove the surrounding noise.

E. More Discussions

Comparison with Non-DL methods. As shown in Figure 9, we further compare our system with non-dl (*i.e.*, non-deep learning based) green screen software DaVinci without human intervention. Our method has better performance under the controlled environment.

To show the effects of these methods in the practical usage scenario, we invite five professionals to interact with these methods' automatically generated results on three groups of videos. We count the time until the professionals are satisfied with the results, and report the average time cost in Table VII. It is found out that by taking our methods as an initialization, combining with interactive methods [77], the professionals make desired videos $10\times$ faster. As the difficulty increases, the time consumption increases, but our method can still make a very quick desired unscreen.

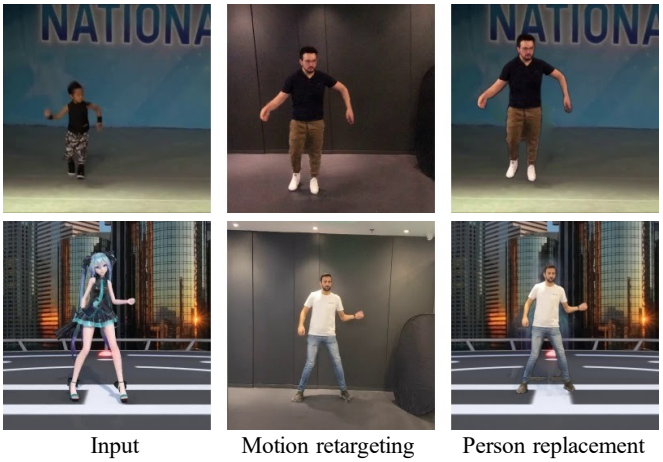

Fig. 10: Video person replacement. The replacement videos have source video's background and target generated foreground. Best viewed in color.

Memory requirements and speed. We test the whole pipeline, including segmentation, matting and background prediction, with a 1080P video on a Titan X GPU. The SOTA method BGM V2 takes $2,281$ MB GPU memory, including models and data, and runs at 2.1 fps. BGM needs $5,308$ MB, 1.0 fps and FBAM uses $5,456$ MB, 0.6 fps. Ours takes $2,130$ MB and achieves 2.3 fps, using the minimum memory while maintaining the fastest speed.

Culprit analysis and ethics. Our method is limited to smooth motion and relatively simple and controlled environments. Although the system is designed to facilitate the art creation and production of more high-quality creative user-generated content, it may also be used as a way to cheat, *e.g.*, to remove the watermark of some commercial videos. [10]

V. Application

The proposed high-performance unscreen system can be applied to human-centric video editing systems in many ways [78]–[81], such as background replacement [82] and people retiming [83]. The key to such applications is a highly accurate separation of foreground and background. For instance, by integrating the estimated backgrounds in our system with motion retargeting [84], [85], an advanced application: *Video person replacement* can be implemented effortlessly.

Existing motion retargeting models generate a fake target video doing the same action as the source video with a different background. However, in their original settings, the target background is fixed to the one in the training data and needs to be static for better performance. Our video unscreen technique makes it possible to replace people in videos with arbitrary backgrounds. As illustrated in Figure 10, we apply our automatic unscreen framework on both the original input video and the motion retargeted video, and separate their backgrounds and alpha foregrounds. To put the

target person at the same position in the source background, we calculate the centroid of the foreground person and acquire the correspondence. With the calculated position correspondence, the target foreground person can be put in the same position in the reconstructed source background. Thus a new video from the alpha foreground of the motion targeting video and the estimated background of the input video is composed. Video demos and more details are put in the supplementary materials.

While generating person replacement videos is of great significance to enrich user-generated content on social platforms, it also bears the risk of the manipulation and the creation of misleading content. Hence, it is of equal importance for researchers to develop methods [86]–[88] that are able to clearly distinguish synthetic contents from real-world contents.

VI. Conclusion

In this work, we propose an automatic background-free trimap-free video unscreen system, which unties the coupling among background and alpha matte, and turns it into a coarse-to-fine refinement process. The system jointly utilizes the semantic and pixel-level information and achieves better performance. We collect a large-scale video mapping dataset *DramaStudio*, comprised of real-world application scenarios. We further show an application on video person replacement built upon our high-quality video unscreen system.

References

[1] "Top 20 ai background remover tools review," https://topten.ai/background-remover-review/, accessed: 2021-05-15. 1

[2] "Unscreen: Remove video background," https://www.unscreen.com, accessed: 2021-05-15. 1, 5

[3] "Remove background from image," https://www.remove.bg, accessed: 2021-05-15. 1, 5

[4] Y. Liu, C. Shen, C. Yu, and J. Wang, "Efficient semantic video segmentation with per-frame inference," 2020. 1, 2

[5] T. Porter and T. Duff, "Compositing digital images," in *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 1984, pp. 253–259. 1

[6] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2970–2979. 1, 2, 3, 4, 5, 6, 7

[7] H. Lu, Y. Dai, C. Shen, and S. Xu, "Indices matter: Learning to index for deep image matting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3266–3275. 1, 2, 5

[8] M. Forte and F. Piti, "F, b, alpha matting," *CoRR*, vol. abs/2003.07711, 2020. 1, 2, 4, 5, 7

[9] S. Sengupta, V. Jayaram, B. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Background matting: The world is your green screen," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2291–2300. 1, 2, 3, 4, 5

[10] S. Lin, A. Ryabtsev, S. Sengupta, B. Curless, S. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," pp. 8762–8771, 2021. 1, 4, 5, 8

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 801–818. 2, 3, 5

[12] B. Zhao, X. Wu, Q. Peng, and S. Yan, "Clothing cosegmentation for shopping images with cluttered background," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1111–1123, 2016. 2

[13] W. Wang and J. Shen, "Higher-order image co-segmentation," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1011–1021, 2016. 2

[14] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji, "Representative discovery of structure cues for weakly-supervised image segmentation," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 470–479, 2013. 2

[15] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1011–1021, 2018. 2

[16] P. Dai, X. Wang, W. Zhang, and J. Chen, "Instance segmentation enabled hybrid data association and discriminative hashing for online multi-object tracking," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1709–1723, 2018. 2

[17] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 267–283. 2

[18] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692. 2

[19] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[20] X. Li and C. Change Loy, "Video object segmentation with joint re-identification and attention-aware mask propagation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 90–105. 2

[21] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6819–6828. 2

[22] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8856–8865. 2

[23] J. Luiten, P. Voigtlaender, and B. Leibe, "Premvos: Proposal-generation, refinement and merging for video object segmentation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 565–580. 2

[24] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "Feelvos: Fast end-to-end embedding learning for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9481–9490. 2

[25] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 5, 7

[26] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7274–7283. 2

[27] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao, "Motion-attentive transition for zero-shot video object segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 066–13 073. 2

[28] Q. Peng and Y.-M. Cheung, "Automatic video object segmentation based on visual and motion saliency," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3083–3094, 2019. 2

[29] B. Jiang, Z. Zhou, X. Wang, J. Tang, and B. Luo, "Cmsalgan: Rgb-d salient object detection with cross-view generative adversarial networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 1343–1353, 2020. 2

[30] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. Torr, "Anchor diffusion for unsupervised video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 931–940. 2

[31] L. Zhang, J. Zhang, Z. Lin, R. Měch, H. Lu, and Y. He, "Unsupervised video object segmentation with joint hotspot tracking," in *Proceedings of the European Conference Computer Vision*. Springer, 2020, pp. 490–506. 2

[32] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan, "Learning discriminative feature with crf for unsupervised video object segmentation," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 445–462. 2

[33] T. Zhou, J. Li, X. Li, and L. Shao, "Target-aware object discovery and association for unsupervised video multi-object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6985–6994. 2

[34] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 7303–7313. 2

[35] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020. 2

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology

for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732. 2

[37] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[38] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 585–601. 2

[39] X. Huang and Y.-J. Zhang, "Fast video saliency detection via maximally stable region motion and object repeatability," *IEEE Transactions on Multimedia*, 2021. 2

[40] Y.-S. Xu, T.-J. Fu, H.-K. Yang, and C.-Y. Lee, "Dynamic video segmentation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6556–6565. 2

[41] S. Jain, X. Wang, and J. E. Gonzalez, "Accel: A corrective fusion network for efficient semantic segmentation on video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8866–8875. 2

[42] C. R. Jung, "Efficient background subtraction and shadow removal for monochromatic video sequences," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 571–577, 2009. 2

[43] S.-Y. Lee, J.-C. Yoon, and I.-K. Lee, "Temporally coherent video matting," *Graphical Models*, vol. 72, no. 3, pp. 25–33, 2010. 2

[44] E. Shahrian, B. Price, S. Cohen, and D. Rajan, "Temporally coherent and spatially accurate video matting," in *Computer Graphics Forum*, vol. 33, no. 2. Wiley Online Library, 2014, pp. 381–390. 2

[45] M. Erofeev, Y. Gitman, D. S. Vatolin, A. Fedorov, and J. Wang, "Perceptually motivated benchmark for video matting." in *Proceedings of the British Machine Vision Conference*, 2015, pp. 99–1. 2, 5

[46] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A bayesian approach to digital matting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2001, pp. II–II. 2

[47] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 315–321. 2

[48] J. Wang and M. F. Cohen, *Image and video matting: a survey*. Now Publishers Inc, 2008. 2

[49] S. Cai, X. Zhang, H. Fan, H. Huang, J. Liu, J. Liu, J. Liu, J. Wang, and J. Sun, "Disentangled image matting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 5

[50] Y. Li and H. Lu, "Natural image matting via guided contextual attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11 450–11 457. 2

[51] Q. Hou and F. Liu, "Context-aware image matting for simultaneous foreground and alpha estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2, 5

[52] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, "Deep automatic portrait matting," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 92–107. 2

[53] B. Zhu, Y. Chen, J. Wang, S. Liu, B. Zhang, and M. Tang, "Fast deep matting for portrait animation on mobile phone," in *Proceedings of the ACM International Conference on Multimedia*, 2017. 2

[54] Y. Zhang, L. Gong, L. Fan, P. Ren, Q. Huang, H. Bao, and W. Xu, "A late fusion cnn for digital matting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[55] Y. Qiao, Y. Liu, X. Yang, D. Zhou, M. Xu, Q. Zhang, and X. Wei, "Attention-guided hierarchical structure aggregation for image matting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020. 2, 5

[56] J. Liu, Y. Yao, W. Hou, M. Cui, X. Xie, C. Zhang, and X.-S. Hua, "Boosting semantic human matting with coarse annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8563–8572. 2, 5

[57] Y. Liu, J. Xie, Y. Qiao, Y. Tang, and X. Yang, "Prior-induced information alignment for image matting," *IEEE Transactions on Multimedia*, 2021. 2

[58] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR.* IEEE, 2004. 2

[59] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 682–695. 2

[60] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Temporally coherent completion of dynamic video," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–11, 2016. 2

[61] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Free-form video inpainting with 3d gated convolution and temporal patchgan," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9066–9075. 2

[62] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5232–5239. 2

[63] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 528–543. 2

[64] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3723–3732. 2

[65] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, "Flow-edge guided video completion," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 713–729. 2, 3, 7

[66] L. Ke, Y.-W. Tai, and C.-K. Tang, "Occlusion-aware video object inpainting," 2021. 2

[67] H. Ouyang, T. Wang, and Q. Chen, "Internal video inpainting by implicit long-range propagation," 2021. 2

[68] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "Temporal video segmentation to scenes using high-level audiovisual features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1163–1177, 2011. 3

[69] A. Rao, J. Wang, L. Xu, X. Jiang, Q. Huang, B. Zhou, and D. Lin, "A unified framework for shot type classification based on subject centric lens," in *Proceedings of the European Conference on Computer Vision*, 2020. 3

[70] X. Jiang, L. Jin, A. Rao, L. Xu, and D. Lin, "Jointly learning the attributes and composition of shots for boundary detection in videos," *IEEE Transactions on Multimedia*, 2021. 3

[71] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 3

[72] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241. 3

[73] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 1999, pp. 246–252. 3

[74] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4471–4480. 3

[75] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 85–100. 3

[76] F. Duo-le and Z. Ming, "A new fast region filling algorithm based on cross searching method," in *Advances in Computer Science and Education Applications*. Springer, 2011, pp. 380–387. 3

[77] Y. Aksoy, T. O. Aydin, M. Pollefeys, and A. Smolić, "Interactive high-quality green-screen keying via color unmixing," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 1, 2016. 8

[78] Y. Chen, D. Bloom, S. Y. Xu, and W. Sun, "Dynamic chroma key for video background replacement," 2019, uS Patent App. 15/945,021. 8

[79] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin, "A local-to-global approach to multi-modal movie scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 146–10 155. 8

[80] Q. Huang, Y. Xiong, A. Rao, J. Wang, and D. Lin, "Movienet: A holistic dataset for movie understanding," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 709–727. 8

[81] J. Xia, A. Rao, Q. Huang, L. Xu, J. Wen, and D. Lin, "Online multi-modal person search in videos," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 174–190. 8

[82] H. Zhang, J. Zhang, F. Perazzi, Z. Lin, and V. M. Patel, "Deep image compositing," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, January 2021, pp. 365–374. 8

[83] E. Lu, F. Cole, T. Dekel, W. Xie, A. Zisserman, D. Salesin, W. T. Freeman, and M. Rubinstein, "Layered neural rendering for retiming people in video," in *SIGGRAPH Asia*, 2020. 8

[84] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5933–5942. 8

[85] Z. Yang, W. Zhu, W. Wu, C. Qian, Q. Zhou, B. Zhou, and C. C. Loy, "Transmomo: Invariance-driven unsupervised video motion retargeting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5306–5315. 8

[86] S. Agarwal and H. Farid, "Detecting deep-fake videos from aural and oral dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 981–989. 9

[87] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of gan-generated fake images over social networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 384–389. 9

[88] J. Sabel and F. Johansson, "On the robustness and generalizability of face synthesis detection methods," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 9