Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A hierarchical general video matting framework towards temporal coherence

Guilin Yao

glyao@hrbcu.edu.cn

Harbin University of Commerce

Shuangyi Cao

Harbin University of Commerce

Xiaowei Han

Harbin University of Commerce

Xu Zhang

Harbin University of Commerce

Article

Keywords: Video matting, spatial-temporal coherence, non-data-driven, Matting Laplacian, color sampling

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11374994/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A hierarchical general video matting framework towards temporal coherence[⋆,⋆⋆]

Guilin Yao[a,*], Shuangyi Cao[a], Xiaowei Han[a] and Xu Zhang[a]

[a]School of Computer and Information Engineering, Harbin University of Commerce, Heilongjiang, 150028, China

ARTICLE INFO

Keywords:
Video matting
spatial-temporal coherence
non-data-driven
Matting Laplacian
color sampling

ABSTRACT

Video Matting is the process of accurately extracting foreground objects from a sequence of input video frames, which requires the accurate estimation not only on the transparency of foreground objects for each frame but also on the spatial-temporal object coherence between consecutive frames. Meanwhile, the prevailing data-driven deep-learning based video matting approaches are constrained by fixed datasets and limited feature aggregation and filtering patterns, making it hard to guarantee the matting results in terms of coherence and accuracy, especially of general foreground objects. Without relying on data-driven patterns, this paper proposes a novel non-data-driven based video matting approach (NDVM) towards general foreground targets and a hierarchical framework towards temporal coherence. Firstly, an KNN-based pre-process layer is employed for adjacent frames according to different foreground and background overlap patterns to classify certain number of opaque pixels. Next, an optical-flow-based sampling layer is adopted for consecutive frames combined with a re-identification criterion for opaque pixels to record both of the transparency and foreground color priors. Finally, an affinity-based smoothing layer based on adaptive kernels is utilized for inter-frames to determine the final video matting results. Experimental results demonstrate that, in terms of quantitative comparison, our approach outperforms both various previous video matting methods and the ablation intermediates, in terms of intra-frame accuracy and inter-frame temporal coherence. Qualitatively, our method also exhibits superior visual perception compared with other state-of-the-art methods. Furthermore, the final foreground results recorded in this paper also provide feasibility and quality assurance for the final video synthesis stage.

1. Introduction

Digital matting is a classic foreground and background separation problem in the field of digital image editing. Unlike image segmentation, real-world images and videos always involve certain translucent and blended pixels at the boundaries or within the foreground objects that require subsequent solution. Therefore, the form of alpha channel is introduced in the process of foreground separation to describe the transparency of the foreground objects. Mathematically Wang and Cohen (2007a), for an observed input frame $\boldsymbol{I}^t$ at time t in a video sequence, the observed color $\boldsymbol{I}^t$ at any position z in an image $\boldsymbol{I}$ can be represented by a linear combination of the foreground color $\boldsymbol{F}_z^t$, background color $\boldsymbol{B}_z^t$, and transparency $\alpha_z^t \in [0, 1]$. The matting equation is then expressed as

$$\boldsymbol{I}_z^t = \alpha_z^t \boldsymbol{F}_z^t + (1 - \alpha_z^t)\boldsymbol{B}_z^t, \tag{1}$$

where the vectors $\boldsymbol{F}_z^t$ and $\boldsymbol{B}_z^t$, and the scalar α_z^t are all unknown. Pixels with $0 < \alpha_z^t < 1$ are referred to as blended or semi-transparent pixels, such as the hair of foreground objects, transparent scarves, and the transition regions of various hard boundaries in Figure 1, and pixels with $\alpha_z^t = 1$ or 0 are referred to as opaque pixels. The matting problem then solves $\{\alpha^t, \boldsymbol{F}^t\}$ corresponding for each frame t given $\boldsymbol{I}^t$ and certain necessary priors. Generally, $\boldsymbol{F}^t$ indicates the color attributions of the foreground object, while α^t indicates the color density distribution and the contour shape of the foreground object. Furthermore, the color distribution of $\boldsymbol{F}^t$ is always continuous, especially in the blended regions where α^t has a complex distribution,

⋆This document is the results of the research project funded by the National Science Foundation.

⋆⋆The second title footnote which is a longer text matter to fill through the whole text width and overflow into another line in the footnotes area of the first page.

This note has no numbers. In this work we demonstrate a_b the formation Y_1 of a new type of polariton on the interface between a cuprous oxide slab and a polystyrene micro-sphere placed on the slab.

*Corresponding author

glyao@hrbcu.edu.cn (G. Yao); caosyvip@163.com (S. Cao); hanxiaowei2017@hrbcu.edu.cn (X. Han); 103044@hrbcu.edu.cn (X. Zhang)

ORCID(s): 0000-0003-0719-4461 (G. Yao)

which makes $\boldsymbol{F}^t$ significantly easier to solve in terms of both demand and difficulty compared to α^t. Only the results of $\{\alpha, \boldsymbol{F}\}$ for a single frame need to be solved in image matting, and it is also necessary to consider the temporal coherence between adjacent matting results α^t of foreground objects in video matting to eliminate discontinuities such as flickering sudden changes.

To reduce the uncertainty, many existing matting methods require a manually annotated mask named trimap to roughly label some known foreground and background regions. However, it can impose a significant input burden on the user for video matting if a trimap is needed for every frame even with a rough annotation manner. In this case, many methods only require to input trimap at certain predefined key frames. Afterwards, the trimap of other frames can be interpolated through adjacent key frames. Additionally, some methods do not require trimaps for each frame and directly deduce the matting result of the current frame only from the results of adjacent key frames, which may introduce more errors in contrast to those with fully provided trimaps.

In image and video matting, similar to other vision tasks, data-driven deep-learning based approaches remain the prevailing methods at present. Despite their numerous notable advantages, deep learning methods cannot fully address the video matting problem, especially towards general foreground targets, or *general video matting*. Currently, data-driven methods still exhibit the following inherent flaws for both image and video matting tasks. Firstly, the matting problem is closed related to the matting equation (1), yet deep-learning based regressions merely approximate pixel probabilities as a proxy for transparency without utilizing matting equation, resulting in biases. Although some methods attempt to compensate the absence of matting equation by the compositional loss, they do not start from the perspective of finding a solution and the results are always unsatisfactory. Secondly, the foreground objects included in various video matting datasets are predominantly hard-boundary objects (such as portraits, animals, rigid objects, etc.), with a lack of diversity in semi-transparent motion, making it hard to fully learn the styles and various motion patterns of various general objects. Thirdly, feature aggregation in deep framework can actually enhance the accuracy of the current frame based on the prior of adjacent frames, but it can hardly solve the temporal coherence between frames. Fourthly, convolution and even transformers are essentially local or global low-pass filtering operations, leading to over-blurred visual senses, particularly in large translucent regions.

Although traditional non-data-driven based processing is inferior to deep-learning in general computer vision tasks, they have obvious advantages in matting, especially in general video matting. This paper primarily explores non-data-driven video matting, which cater to various general foreground objects and strive to avoid being constrained by the semantic branches of limited foreground object training datasets. Our approach does not rely on convolutions that can lead to over blurred results. Instead, it directly extracts samples from the original image and utilizes the matting equation to obtain an accurate solution for α transparency. Meanwhile, our approach directly enhances the temporal coherence through samples and affinities between adjacent frames, avoiding the obscure feature aggregation in deep frameworks. Extensive experiments show that this non-data-driven based video matting results are significantly superior to those based on the same category and even deep learning in terms of coherence and accuracy.

2. Related work

Traditional non-data-driven image matting can be divided into sampling-based Chuang, Curless, Salesin and Szeliski (2001); Wang and Cohen (2007b); He, Rhemann, Rother, Tang and Sun (2011); Gastal and Oliveira (2010) and affinity-based Levin, Lischinski and Weiss (2008); Chen, Li and Tang (2013); Aksoy, Aydun and Pollefeys (2017) methods. Due to the reliance on low-level features and exclusion of any semantics, these methods are characterized by the necessity of using trimaps to guide the rough semantic segmentation. In contrast, data-driven based deep image matting utilizes an encoder-decoder network to extract semantics, allowing for both general object matting that utilizes a trimap to enhance foreground semantics Xu, Price and Cohen (2017); Lu, Dai, Shen and Xu (2019); Cai, Zhang, Fan, Huang, Liu, Liu, Liu, Wang and Sun (2019); Li and Lu (2020); Park, Son and Yoo (2022); Yao, Wang, Yang and Wang (2024a); Hu, Lin, Wang, Zhao, Wei and Shi (2024) and specific foreground object (such as portrait) matting that does not require any trimap Shen, Tao, Gao, Zhou and Jia (2016); Chen, Ge, Xu, Zhang, Yang and Gai (2018); Liu, Yao and Hou (2020); Li, Zhang, Maybank and Tao (2022); Yao, Wang, Ye and Liu (2024b).

Video matting not only requires the accuracy for a single frame, but also ensures the spatial-temporal consistency of α between adjacent frames. According to the form of motion of the foreground object, α results at the same position of the corresponding motion trajectory between adjacent frames should be as consistent as possible, so that the matting result could not visually exhibit obvious discontinuities such as flickering and jumping.

2.1. Non-data-driven based video matting

Similar to traditional non-data-driven image matting, a notable feature of non-data-driven video matting approaches is still the necessity of trimap, but the difference among various sub-methods lies in whether the trimap is required for all frames. Based on the input forms of the trimap, traditional approaches can be divided into methods based on full-frame provided trimaps and those based on keyframe trimaps. [1] provides a partial overview of traditional video matting. Another feature is that most methods simply compare α while ignoring the computation of foreground color $\boldsymbol{F}$, which is not conducive to the level of final video synthesis applications.

Full-frame based trimap input methods. Interactive video cutout Wang, Bhat, Colburn, Agrawala and Cohen (2005) treats all image frames as a single 3D processing target and adopts an interactive approach to provide initial rough trimaps for all input frames. It then utilizes graph cuts to uniformly compute binary segmentation results for each frame to form a refined trimap and finally employs 3D border matting to process the matting of consecutive frames. Temporally coherent video matting Lee, Yoon and Lee (2010), CWCT video matting Shahrian, Price, Cohen and Rajan (2012), and KL divergence video matting Karacan, Erdem and Erdem (2017) adopt their respective sampling methods for each frame, while utilizing optical flow to calculate the positional correspondence between two frames to expand the samples of the current frame. Furthermore, method in [23] also selects appropriate samples for the current frame based on the compatibility of adjacent α. All of the three methods also establish a 3D Matting Laplacian to post-process the consecutive 2 frames according to the foreground motion to maintain the temporal consistency. The affinity-based non-local video matting method Choi, Lee and Tai (2012) employs a non-local Laplacian video matting matrix defined in temporal space for 3 consecutive frames and uniformly solves these frames.

Keyframe based trimap input methods. Some early methods first employ binary segmentation and dilate the segmentation boundary to form a fine trimap followed by matting. Video Cut and paste Li, Sun and Shum (2005) and Video Snapcut Bai, Wang, Simons and Sapiro (2009) can only handle relatively simple foreground boundaries (i.e., hard boundaries) and are unable to handle foreground edges with abundant hair and translucent objects (such as plastic or scarves). Additionally, they can only handle α and then ignore foreground color. Later, new methods can calculate complex boundaries and translucent objects, which are common in affinity-based video matting methods. Alpha-Flow video matting Sindeev, Konushin and Rother (2012) simply uses optical flow to propagate the prior results of Robust Matting [3] to the next frame. However, it does not consider the joint solution of adjacent frames, which may result in discontinuities in the matting results. The opacity propagation video matting Tang, Miao, Wan and Zhang (2012), KNN video matting Li, Chen and Tang (2013), sparse coding video matting Zou, Chen, Cao and Wang (2020) all utilize the foreground and background result pixels from previous frames as soft constraints to jointly solve for the current frame and the previous frame, and propagate the results backward frame by frame until the next keyframe. This enhances the result coherence between adjacent frames, but the unidirectional propagation can lead to discontinuities when propagating to the next keyframe. Additionally, none of these methods can solve foreground color. The PatchMatch-based matte propagation method Backes and Oliveira (2019) improves the above methods to a bidirectional propagation fashion. It first calculates the results of keyframes by some single-frame matting method, and then propagates $\{\alpha, \boldsymbol{F}\}$ from the keyframe to unknown frames in a bidirectional manner by optical flow based on PatchMatch algorithm and cyclic filtering.

It has been proven both theoretically and practically that the initial trimap input is crucial for single-frame solution. Therefore, methods based on full frames are significantly superior to those based on key frames. However, as discussed, the main issue with such methods is the need to provide a sufficient dataset of trimaps, and manually inputting trimaps for all frames is extremely difficult. Consequently, a class of video trimap propagation methods has emerged to address this issue. These methods extend the trimap at key frames to non-key frames and still require full-frame matting methods in the subsequent steps. Geodesic video matting Bai and Sapiro (2009) adopts a weighted geodesic distance approach, while Bayesian matting Chuang, Agarwala, Curless, Salesin and Szeliski (2002) employs optical flow and a combination of forward and backward propagation to extend the precise trimap of key frames to unknown non-key frames. Temporally coherent video matting Bai, Wang and Simons (2011) utilizes user-input trimaps of key frames to calculate the trimap of unknown frames by combining optical flow with backward propagation. Temporal trimap extension Johnson, Rajan and Cholakkal (2015) extends the fine trimap from key frames to other frames based on the contours of foreground objects combined with optical flow, which can only be applied to hard boundary contours.

Similar to image matting, the result of each frame in general video matting still heavily relies on the trimap input. As discussed above, many of the non-data-driven video matting methods still directly employ trimap only at key frames and introduce pure alpha propagation to obtain the entire final alpha matting results, which will inevitably cause inaccurate propagations, especially in

mixed regions where fine matting is required. As shown in Figure 1, our method requires a trimap input for each frame, simultaneously solves for $\{\alpha, \boldsymbol{F}\}$ of adjacent 2 frames to achieve inter-frame consistency, and ultimately achieves the goal of synthesizing the pulled mattes with new backgrounds. To reduce input burden, users are allowed to annotate a video sequence with sparse trimaps at key frames, and then a simple trimap extension step is employed to annotate all the other frames with roughly coarse trimaps. Unlike propagating α, the coarse trimap propagation basically could not cause significant errors.

2.2. Data-driven based video matting

Currently, data-driven based deep video matting still employs the encoder-decoder network from deep image matting for each frame. The first category involves portrait video matting that has clear portrait semantics and does not require initial trimap input. Both Background Video Matting Sengupta, Jayaram, Curless, Seitz and Kemelmacher-Shlizerman (2020) and Real-Time High-Resolution Background Video Matting Lin, Ryabtsev and Sengupta (2021) require capturing a single fixed background and then perform matting for all the foreground individuals appearing in front of this background. MODNet Ke, Sun and Li (2022) also proposes a dual-encoder-decoder scheme similar to image matting. However, the above methods only consider frame-by-frame matting and ignore the inter-frame correlations. Robust High-Resolution Video Matting Lin, Yang and Saleemi (2022), Automatic Portrait Video Matting Hou and Wang (2021), SparseMat Sun, Tang and Tai (2023), VideoMatt Li, Ohanyan, Goel, Navasardyan, Wei and Shi (2023), and VMFormer Li, Goel, Ohanyan, Navasardyan, Wei and Shi (2024) all adopt inter-frame feature aggregation to handle the relevance problem between frames.

The second category is for video matting of general foregrounds. At this time, due to the overly rich and difficult-to-generalize foreground semantics, auxiliary trimaps are required frame by frame. The first basic pattern, including DVM Sun, Wang and Gu (2021), TCVOM Zhang, Wang and Cui (2021), and CRGNN Wang, Liu and Tian (2021), utilizes the user-input trimap of key frames, passes it to other frames through a spatial-temporal memory network (STM), and then utilizes the inter-frame feature fusion between adjacent frames and attention mechanisms to enhance the correlation between frames. The second pattern is to utilize the trimap at key frames, combined with the STM segmentation network, to process the generation of trimaps for other frames in parallel with the final matting. FTP-VM Huang and Lee (2023) changes the memory matching in STM to a lightweight trimap fusion module, which greatly improves the matting efficiency. OTVM Seong, Oh, Price, Kim and Lee (2022) draws upon the conception of [35] by using previously solved α and the trimap of the initial frame to generate the trimap of new frames.

3. Our approach

The hierarchical video matting framework presented in this paper is illustrated in Figure 1. The first affinity-based pre-processing layer (Section 3.1) fully leverages the ability of affinity-based category to capture long-distance dependencies, and pre-identifies some unknown pixels in two consecutive frames as opaque pixels, laying the groundwork for subsequent matting layers. The second sampling layer (Section 3.2) fully utilizes the sampling-based category in accurately handling individual pixels by solving intermediate α results and the final foreground color $\boldsymbol{F}$ and the corresponding beliefs for the current two frames. The third affinity-based solver layer (Section 3.3) utilizes various intermediate results from the previous two layers to solve the final α for the current two frames.

3.1. Pre-processing layer

In image and video matting, performing preliminary trimap expansion on each frame is crucial for the final matting result. This step enables more precise pre-prediction of opaque pixels, while providing more accurate samples for subsequent precise matting steps. However, most existing video matting algorithms still directly employ matting, with few variants incorporating a trimap expansion step beforehand. Up to now, nonlocal-based algorithms excel in image segmentation on opaque pixels (i.e., trimap expansion or generation in matting) due to their ability to search for distant samples. This principle still holds in video matting, and such search needs to be conducted across frames to reflect frame correlations.

As shown in Figure 1, the first layer employs a strategy similar in [12], the affinity-based intra-frame and inter-frame K-nearest neighbor search, to roughly estimate the foreground probability of each frame in order to enhance temporal coherence across frames. The basic assumption is that the foreground prob of each unknown pixel in the current frame t is a linear combination of the foreground probabilities $p_1^t,\ p_2^t, \ldots, p_K^t$ and $p_1^{t+1},\ p_2^{t+1}, \ldots, p_K^{t+1}$ of the K nearest neighbors in frame t and $t+1$ respectively. In KNN search, samples at certain distances can be searched by scaling the distance vectors. The feature vector of pixel i is defined as $\boldsymbol{v}_i^t(f) = \left[r_i, g_i, b_i, f\,x_i, f\,y_i\right]$, where $\left\{r_i, g_i, b_i\right\}$ represent the colors of the three channels of pixel i in frame t, x_i and y_i are the normalized spatial horizontal and vertical coordinates of point i, and f is the spatial scaling factor. The larger f is, the closer the collected pixels are, and vice versa. Then a FLANN fast

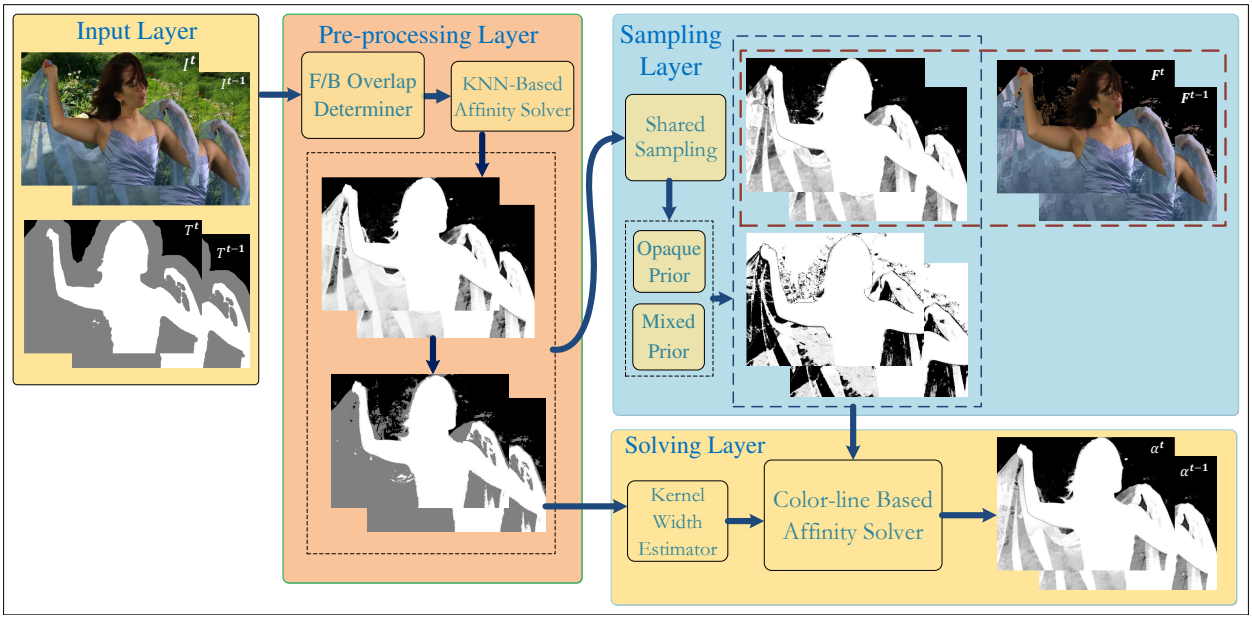


Figure 1: The proposed hierarchical video matting framework.

search is adopted to match K neighboring pixels at time t for both frame t and frame $t+1$.

Whether in single or multi-frame mode, the value of f in various nonlocal methods mentioned in the literature has not been precisely defined and requires manual adjustment, ultimately resorting to a fixed search range to limit the collected nearby pixels. However, in practice, it has been proved that the search range must vary depending on the degree of overlap between foreground and background colors. Generally, if there is a significant color difference between the foreground and background or a large size of the unknown region, a relatively loose search can be accepted so as not to miss more distant reference samples. Otherwise, a strict search is necessary to avoid erroneous caused by color overlaps. Hence, the overlap degree between F&B colors as well as the size of the unknown regions are both precomputed. Here, two narrow bands within the known F&B region with a width of 30 pixels adjacent to the unknown region Ω_U are denoted as Ω_F and Ω_B. Then, the initial FLANN mappings between the source and the target region are denoted as $\Omega_{\mathrm{F}} \rightarrow \Omega_{\mathrm{B}}$ and $\Omega_{\mathrm{B}} \rightarrow \Omega_{\mathrm{F}}$ based only on the first three color features of $v_i^t(f)$. In the first frame, the average color differences $\bar{d}(\mathrm{F},\ \mathrm{B})$ and $\bar{d}(\mathrm{B},\ \mathrm{F})$ are computed respectively with $K = 5$. If the minimum of these two differences is greater than a certain threshold θ_1, and the average spatial distance between all the unknown pixels and the Ω_{F} and Ω_{B} regions is greater than a second threshold θ_2, there should be a high F&B distinguishability or a large size of the unknown region. In this case, a much loose search is needed; otherwise, a relatively strict search should be considered instead.

In the loose search, the distance scale f of the final FLANN search $\Omega_{\mathrm{U}} \rightarrow \Omega_{\mathrm{U}} \cup \Omega_{\mathrm{F}} \cup \Omega_{\mathrm{B}}$ can be divided into three levels $\{1, 0.1, 0.01\}$. Within these ranges, $\{5,\ 8,\ 8\}$ nearest neighbor samples j are collected for pixel i in frames t and $t+1$, respectively. In the strict search, the search range is reduced to two levels with distance scale $\{1, 0.1\}$ and sample count $\{8, 5\}$. Additionally, it is observed that if $\bar{d}(\mathrm{B},\mathrm{F})$ is greater than a certain threshold θ_3, indicating a significant distinction between F&B, HSV color components for $v_i^t(f)$ could yield much better results under the loose search. Here $\{\theta_1, \theta_2, \theta_3\} = \{0.05, 50, 0.2\}$. Figure 2 shows the pre-processing results of four frames from four video sequences. Among them, the second and third examples have a high F&B overlap between and are processed with the strict search. The first (using HSV color space) and the fourth examples have a lower F&B overlap and are processed with the loose search. It is clear that a proper search could pick out more opaque pixels from the unknown region and avoid misclassification to a certain extent.

Finally, the affinity-based sparse matrix L is constructed from the neighboring samples, whose length and width are both twice of the entire frame pixel number N. Specifically, the intra-frame and the inter-frame affinity matrices with respect to the first frame can be calculated separately, denoted as L_{11} and L_{12}, respectively. Similarly, the affinity matrices corresponding to the second frame are denoted as L_{21} and L_{22}. Let $L_1 = \left[L_{11},\ L_{12}\right]$ and



Figure 2: Trimap pre-processing results obtained using the loose search (second row) and strict search (third row) for four specific video frames with initial trimaps (first row) in four video sequences. The red box indicates the optimal selection.

$L_2 = [L_{21},\ L_{22}]$, and the final L for the two consecutive frames is generated as $L = [L_1; L_2]_{2N\times 2N}$. After obtaining sample j for unknown pixel i, the entry in the i^{th} row and j^{th} column of the sparse matrix L is calculated as

$$w_{ij} = 1 - \sum_{c\leq 3} \left|c_i^c - c_j^c\right| / 3. \tag{2}$$

The final foreground probability (or pre-processing result) α^{pre} is solved by the following sparse linear system

$$(L + \theta D)\,\alpha^{\text{pre}} = \theta b, \tag{3}$$

where D is a $2N \times 2N$ diagonal matrix, with 1s at the entry of known regions $\Omega_{\text{F}} \cup \Omega_{\text{B}}$ and 0s at the entry of unknown region Ω_{U} of the 2 adjacent frames. θ is some large number, and b is a column vector with a length of 2N, with 1s at Ω_{F}, and 0s at $\Omega_{\text{B}} \cup \Omega_{\text{U}}$. The above system is solved by conjugate gradient approach for α^{pre}. Finally, the new opaque foreground and background regions of the adjacent 2 frames are the pixels i that satisfy $\alpha^{\text{pre}} > 0.985$ and $\alpha^{\text{pre}} < 0.015$, respectively. The above new regions as well as the new unknown region constitute the final refined 2-frame trimap, shown in the pre-processing layer in Figure 1.

3.2. Sampling layer

The sampling layer requires full utilization of the trimap results from the previous layer, and provides priors for the final affinity-based matting layer. Meanwhile, this layer can also preserve the foreground colors in preparation for the final video synthesis, overcoming the drawback of affinity-based methods that can only solve for α once without simultaneous foreground computation. Besides, the samples from the previous frame could compensate for the current frame, thus fully integrating the correlation between both frames and also compensating for the lack of inter-frame consistency in affinity-based methods.

The sampling method draws inspiration from the shared matting approach. The candidate samples for the current unknown pixel need to be collected from the primary candidate samples of neighboring unknown pixels, thus avoiding the excessive complexity in traditional sampling mode where samples are enumerated every pair. In the first **sample collection** step, each unknown pixel u in frame t emits only four rays with 90° intervals, intersecting with the initial input F&B regions $\Omega_{\text{F}}/\Omega_{\text{B}}$, as well as the F&B regions after trimap expansion $\Omega'_{\text{F}}/\Omega'_{\text{B}}$. Then a total of up to 8 F&B samples are collected, resulting in a maximum of 8×8 sample pairs. Figure 3(a) shows the samples collected for an unknown pixel. For each sample pair F^j, B^k at point u, the corresponding α_u is inversely solved according to the matting equation (1), as well as the local color fitting error $N_u\left(F^j, B^k\right)$ within a 3×3 neighborhood. Then the spatial shortest distances between u and foreground sample F^j or background sample B^k are denoted as $D_u\left(F^j\right)$ and $D_u\left(B^k\right)$, respectively. Then the objective function for the initial selection of samples at each unknown point is defined as

$$g_u\left(F^j, B^k\right) = N_u\left(F^j, B^k\right)^s \cdot D_u\left(B^k\right)^{t_1} \cdot D_u\left(B^k\right)^{t_2}, \tag{4}$$

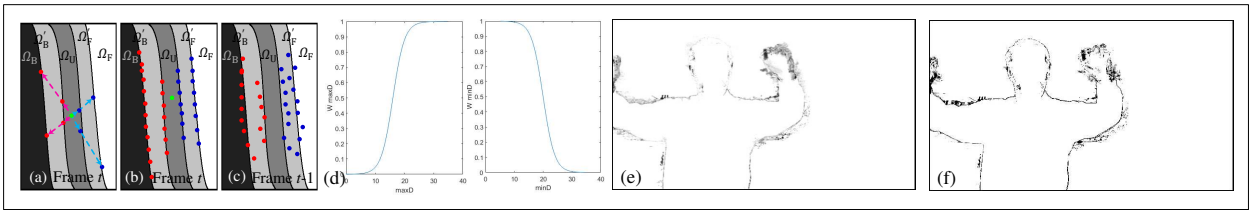


Figure 3: (a) The four sampling rays for each unknown pixel u in the sample collection step, which intersect with the edges of both the input trimap and the newly generated trimap. (b) All T_u foreground and background samples for unknown pixel u in the current frame t in the neighbor searching step. (c) All samples of u in frame $t-1$. (d) Curves of maximum and minimum color difference weights. (e) Belief map in unary term of color fitting error from the third example of Fig. 2. (f) Final belief map with the addition of maximum/minimum color difference weights.

where $\{s, t_1, t_2\} = \{3, 1, 4\}$. Since the unknown regions, such as hair or hollow, usually appear far from known background and much closer to foreground, the distance of background samples is set to a relatively large cost to remove distant background samples. Finally, the optimal sample pair (F_u, B_u) for pixel u is determined by the pair with the minimum objective function g_i and is recorded for u in frame t. In the next **neighbor searching** step, the candidate samples of each unknown pixel i are collected from the stored samples (F_i, B_i) of its $T_u = 200$ neighborhood. Although the sampling rays are relatively sparse, the amount of densely packed T_u neighbors ensure dense sample collection at the boundary. Optical flow is also applied here to compensate the temporal coherence only for the latter frame in the trimap expansion step. A widely concept is that optical flow has poor performance in transmitting mixed pixels. However, it is effective on opaque pixels such as samples. Therefore, each sample pair (F_i, B_i) in the current frame also needs to be warped to the previous frame to form a new warped sample pair $\left(F_i^{\text{opt}}, B_i^{\text{opt}}\right)$. Here, all previous frame samples are required to be inside the opaque region $\Omega'_{\text{F}}/\Omega'_{\text{B}}$. Figures 3(b)(c) show the samples of unknown point u in the current frame and the previous frame. Finally, the 10 sample pairs (F_i^r, B_i^r) that minimize N_i are selected, and the primary sampling α'_i of unknown pixel i is obtained by solving equation (1), with F'_i and B'_i being the average of all the samples.

Opaque prior. The color criterion involves the color difference between unknown and known pixels. The classification threshold for each known opaque pixel is obtained according to its color difference to neighboring pixels, and is stored for the classification of all unknown pixels. Compared to fixed threshold, this dynamic way could significantly improve the classification performance. Taking opaque foreground classification as an example. For any opaque foreground pixel F_i in the new foreground region Ω'_{F}, the average color difference between F_i and its local neighborhood centered on F_i with a radius of 6 is recorded in F_i, requiring that the neighboring pixels are also in Ω'_{F}. Then for any unknown pixel i, the top 20 samples F_i with the smallest color difference among its T_i foreground samples are obtained, and the color differences and thresholds recorded by these samples are averaged, denoted as $\bar{D}_i^{\text{F}}$ and $\bar{\sigma}_{\text{F}}$. Similarly, the background color differences and thresholds are denoted as $\bar{D}_i^{\text{B}}$ and $\bar{\sigma}_{\text{B}}$. The color criterion for classifying pixel i as opaque foreground is defined as

$$\bar{D}_i^{\text{F}} \le \bar{\sigma}_{\text{F}} \,\&\, \bar{D}_i^{\text{B}} > \bar{\sigma}_{\text{B}} \,\&(\, \alpha_i^{\text{pre}} > 0.9 | \alpha'_i > 0.99), \quad (5)$$

where α_i^{pre} and α'_i denote the pre-processing and sampling α results. The color criterion for determining pixel i as an opaque background can be derived similarly.

The distance criterion involving the spatial distance between the unknown pixel and the known boundary can also compensate color. Through the ray-casting in sec. 3.1, the minimum E_i^{F} of the cumulative color difference from the paths traveled by each ray to the foreground, and the minimum value E_i^{B} to the background are both computed for pixel i. Denoting $d_i^{\text{F}} = E_i^{\text{B}}/(E_i^{\text{B}} + E_i^{\text{F}})$, the distance criterion for determining pixel i as an opaque foreground is defined as

$$d_i^{\text{F}} > 0.98 \,\&\, \alpha_i^{\text{pre}} > 0.6. \quad (6)$$

The distance criterion can also be defined for background similarly. The first row of Figure 4 shows the four examples of opaque priors from Figure 2. Finally, if i is classified as foreground, its prior $\alpha_i^{\text{samp}} = 1$, $F_i^{\text{samp}} = I_i$, and $B_i^{\text{samp}} = B'_i$. If i is classified as the background, $\alpha_i^{\text{samp}} = 0$, $F_i^{\text{samp}} = F'_i$, and $B_i^{\text{samp}} = I_i$. Its belief is uniformly set to $f_i = 5$. Since the final affinity-based matting layer cannot solve the foreground and background colors, and α always exhibit little color variation spatially, the foreground and background color results are thus saved as the final video matting results.

Mixed prior. If i is not classified as opaque, it is treated as mixed with $\alpha_i^{\text{samp}} = \alpha'_i$, $F_i^{\text{samp}} = F'_i$, $B_i^{\text{samp}} =$

B_i', and its belief f_i^r is calculated as

$$f_i = W_i^{\text{maxD}} \cdot W_i^{\text{minD}} \cdot \left\{-\lambda \bar{N}_i\right\}, \tag{7}$$

where $\lambda = 0.05$ and $\bar{N}_i$ represents the average of the top 10 smallest N_i, and the weights of the 2 maximum/minimum color differences are defined as

$$\begin{cases} W_i^{\text{maxD}} = (1 + \exp(-(\max(\bar{D}_i^{\text{F}}, \bar{D}_i^{\text{B}}) * 0.5 - 8)))^{-1} \\ W_i^{\text{minD}} = (1 + \exp(-(-\min(\bar{D}_i^{\text{F}}, \bar{D}_i^{\text{B}}) * 0.5 + 10)))^{-1} \end{cases} \tag{8}$$

where $\bar{D}_i^{\text{F}}$ and $\bar{D}_i^{\text{B}}$ represent the average color differences between unknown pixel i and the foreground and background, respectively. Figure 3(d) shows the curves for the above two weights. Specifically, the belief needs to be punished if the maximum/ minimum color difference is less/greater than a certain range. Figure 3(e) and Figure 3(f) shows the belief map by only applying the color fitting error $\bar{N}_i$ and the final belief map in addition of the above two color difference weights, respectively. The second and third rows of Figure 4 show the final belief maps f and the final α^{samp} obtained through the sampling layer for the four examples in Figure 2.

3.3. Matting layer

Although sampling-based processes can trace the results of foreground color, they tend to produce visually noisy results. On the other hand, affinity-based processes could generally yield better results in terms of visual smoothness and comfort. Therefore, the third layer combines the sampling-based results from the above layer and a newly designed affinity-based post-processor.

As discussed, a fine matting step requires the weight type of Matting Laplacian [3,11] utilizing matting equation rather than simple nonlocal weight. Meanwhile, since the resultant fine trimap can roughly reflect the foreground shape, different forms of Matting Laplacian kernels can be appropriately adopted according to different sizes of the unknown region, thus satisfying different local color line model assumptions. In traditional, both 2D and 3D Matting Laplacian typically employ a fixed local neighborhood window that always leads to significant color differences and violates the assumption of the color linear model. In our approach, a means of color feature matching instead of spatial distance matching is adopted to discover long range neighboring samples to enhance pixel dependency.

After the above steps, a considerable number of opaque pixels have been identified as known, and the foreground contours in the remaining unknown region have also been roughly outlined, including hard boundaries, soft boundaries, and the interior of semi-transparent objects. Accordingly, the selection of the search range in color matching can also play an important auxiliary role, and the generation of the neighborhood set in 3D Matting Laplacian also requires the selection of an appropriate search range. On the one hand, the search range in each frame should be as close as possible to the foreground and background regions in the solved trimap, in order to directly obtain known samples in a similar color sampling manner and avoid excessive transfer between intermediate pixels; on the other hand, the search range should also be properly selected in order to reduce computational costs and minimize errors caused by color overlap between foreground and background.

Specifically, in the current frame t, with the previously newly generated trimap, the shortest spatial distances between each unknown pixel i and the new known foreground Ω_{F}' and background Ω_{B}' are updated, denoted as D_i^{F} and D_i^{B}, and with their average denoted as $\bar{D}_i^t$. The 4th row of Figure 4 shows the average spatial distances $\bar{D}^t$ of each unknown pixel in the four examples. Subsequently, the search range for color matching is determined based on $\bar{D}^t$ according to the following two cases. (i) $\bar{D}_i^t \leq 6$ indicating i is close to a hard boundary, and a square search window is required trying to cover both known foreground and background regions, but also avoiding too much size. The search radius r is defined as

$$r = \max\left(3, \min\left(\max\left(D_i^{\text{F}},\ D_i^{\text{B}}\right),\ 9\right)\right). \tag{9}$$

Within the defined r, 10 (totaling 20) nearest neighbors of pixel i in frame t and $t+1$ are searched based on the 3 color features. For accuracy, the position of i in frame $t+1$ can be appropriately adjusted by optical flow. (b) $\bar{D}_i^t > 6$ indicating i is close to the soft boundary or the interior of a semi-transparent object. Similar to Sec. 3.1, FLANN search is applied with $f = 0.5$ and $f = 0.1$ in the feature vector $v_i^t(f)$ respectively to discover $\{10, 2\}$ nearest neighbors (totaling 24) of pixel i in the current frame t and $t+1$.

In the above two cases, each unknown pixel i in frame $t+1$ also performs similarly to discover its corresponding neighboring pixels in these two frames. Finally, each i forms the neighborhood set S in the 3D Matting Laplacian. Note that set S needs to be determined in the adjacent two frames. By utilizing the color line model and combining the derivation of the matting equation, any two pixels $(i, j) \in S$ within set S could constitute the weights of the i-th row and j-th column in the sparse matrix L as

$$w_{ij} = -\frac{1}{|S|}\left(\mathbf{1} + \left(\boldsymbol{x}_i - \boldsymbol{\mu}_S\right)\left(\Sigma_S + \frac{\varepsilon}{|S|}X_3\right)^{-1}\left(\boldsymbol{x}_j - \boldsymbol{\mu}_S\right)\right), \tag{10}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ represent the 3-dimensional color vectors of pixel i and j, respectively, and X_3 denotes a 3×3

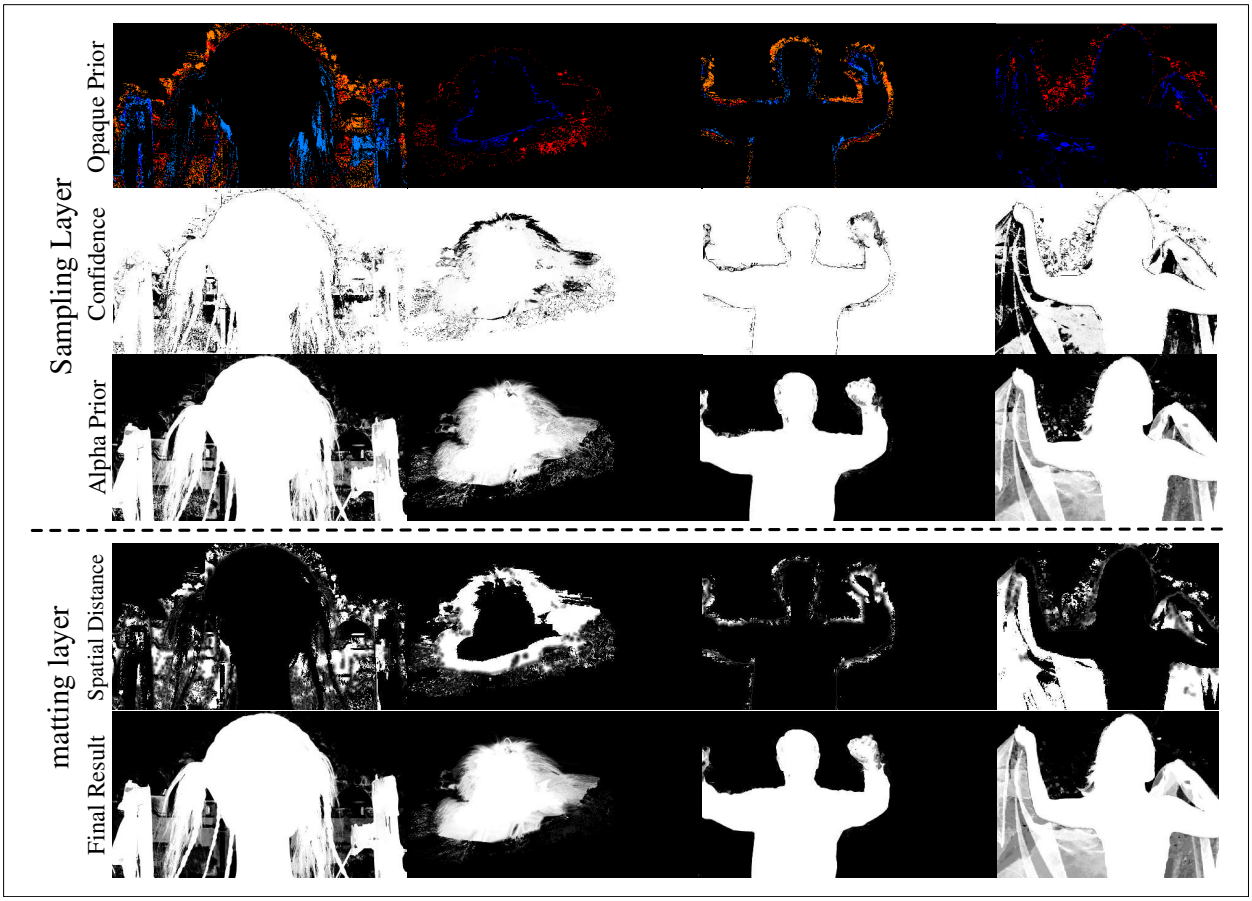


Figure 4: Intermediate and final results of the second and third layers for specific video frames in the four video sequences in Fig. 2.

identity matrix. Σ_S and $\boldsymbol{\mu}_S$ stand for the 3×3 covariance matrix and the 3×1 mean vector within S, respectively, with ε being a small number. The weight of unknown pixels i and j is the sum of weights generated by all sets S containing them. Combining the prior and belief obtained from the sampling layer, the α results for two consecutive frames can be obtained by solving the following large sparse linear equation

$$(L + \theta D + \lambda \cdot \mathrm{diag}\,(1))\,\alpha = \theta b + \lambda \cdot \alpha_i^{\mathrm{samp}} \qquad (11)$$

where D, θ, and b represent the hard constraints of Ω'_{F} and Ω'_{B} after the trimap expansion layer for adjacent 2 frames, and are set similarly to Sec. 3.1. α_i^{samp} is the prior of the sampling layer in Sec. 3.1, diag is a function that expands a column vector into a diagonal matrix, and $\lambda_i = f_i \cdot d_i \cdot \gamma$ is the weight of the data term where f_i is the sampling belief, $\gamma = 0.1$, and d_i is a distance weight calculated as

$$d_i = 1 + 9 \cdot \left(1 + \exp\left(-\left(\bar{D}_i^t \times 0.4 - 12\right)\right)\right)^{-1} \qquad (12)$$

where $\bar{D}_i^t$ represents the average distance from pixel i to the known region as previously computed. d_i indicates that the farther an unknown point is from the known boundary, the less reliable the affinity-based method is. In such case, the pixel needs to rely more on the prior from sampling layer. The final α results obtained from the eq. 11 are shown in the last row of Figure 4.

4. Experiments

4.1. Dataset and Methods

In the experiments, the first dataset is sourced from the 5 sets of human foreground sequences from the validation set of the VideoMatte240 dataset [38], which contains ground truths. These sequences are synthesized into 5 different dynamic backgrounds derived from DVM [45], resulting in a total of 25 new test video sets. In order for this general input trimaps and the solvability of our approach, we introduce 2 types of trimaps, referred as "large" and "small" obtained by dilating the unknown

regions in the ground truth α by 30 and 60 pixels, respectively. To demonstrate inclusivity towards various datasets, we simultaneously introduce a second Video Matting Benchmark Dataset Erofeev, Gitman, Vatolin, Fedorov and Wang (2015) (VM) comprising 16 training and testing videos with featuring various foregrounds. The original VM dataset consists of only three training sets containing ground truths. In our work, some of the 5 test sequences included in the Hollywood Green Screen Dataset were re-matted using Adobe After Effects to obtain their ground truth values and added to our training set. Then, our second video matting dataset comprises 8 training sets and 8 test sets. Besides, the trimap generation manner for this second dataset is the same as that of the first one. Consequently, the entire 33 training video sets and 8 test video sets are all used for testing in the experiments.

In the experiments, the non-data-driven hierarchical video matting presented in this paper is denoted as NHVM, and another 7 non-data-driven video matting methods are also included. To demonstrate the superiority of dual-frame over single-frame, we first introduce the traditional single-frame robust matting [3] for frame-by-frame processing, and then adopt the temporal filter based on level set [35] on robust matting. The following three methods are dual-frame processed with 3D Matting Laplacian, including the benchmark dual-frame closed-form matting (CLF3D), dual-frame 3D robust sampling [22] and sparse coding [31] (Robust3D and Sparse3D), respectively. After that, there are two non-local methods, including dual-frame nonlocal matting [25] (Nonlocal) and KNN Laplacian matting [30] (KNNLap), both of which apply 2-frame trimap input instead of trimap inputs at key frames. Note that CLF3D itself does not appear in any proposed method, but is implemented here for comparison. For comparison, we also introduce 4 data-driven based general deep video matting methods, including DVM [45], TCVOM [46], CRGNN [47], and FTP-VM [48], all of which require a trimap for each frame. Note that frame-by-frame processing is adopted in FTP-VM due to its parallel computation of trimap and α in multi frames.

4.2. Quantitative comparison

For comparison, the following two evaluation metrics are included denoted as intra-frame accuracy and the inter-frame coherence (or continuity). For each video matting method, under specific video sequences and trimap models, the intra-frame mean absolute difference (MAD), root mean square difference (RMSD) [50] and the inter-frame temporal coherence (AdtIdt) [22] over the entire dataset are defined as

$$\begin{cases} \mathrm{MAD}\,(\alpha) = \frac{1}{N}\sum_t \sum_{i\in S} \left|\alpha_i - \alpha_i^{\mathrm{true}}\right| \\ \mathrm{RMSD}\,(\alpha) = \sqrt{\frac{1}{N}\sum_t \sum_{i\in S} \left(\alpha_i - \alpha_i^{\mathrm{true}}\right)^2} \\ \mathrm{AdtIdt}\,(\alpha) = \frac{1}{N}\sum_t \sum_{i\in S} \frac{\alpha_i(t+1)-\alpha_i(t)}{I_i(t+1)-I_i(t)} \end{cases}, \quad (13)$$

where N denotes the total number of pixels in the unknown region corresponding to the video sequence and trimap, t denotes the frame index, i denotes the pixel index, and α and α^{true} signify the α result and ground truth α obtained from each frame in each sequence, respectively. t and $t+1$ indicate the current and the next frame, respectively, and S denotes a specific set, such as the unknown or the known region. A smaller value for both evaluation metrics indicates a better matting result.

4.3. Experimental Realizations

4.3.1. Macro comparison for all video sequences

To demonstrate the macro effects of various methods on different pixel regions, the opaque region and the mixed region are firstly selected corresponding to S in Eq. 13, where the opaque region is defined as those pixels with α^{true} being greater than 0.98 or less than 0.02, and the mixed region corresponds α^{true} between these two values. The comparison of various methods, including our first layer (pre-processing), second layer (sampling) and final layer (affinity-based solver), under evaluation metrics RMSD and AdtIdt with different trimap sizes in the entire dataset is shown in Tables 1. Each data for MAD and RMSD is an average of 33 training video sets including 25 test sequences from VideoMatte240 and 8 training sequences from VM-Benchmark, while AdtIdt is an average of all 41 video sets including 8 extra test sequences in VM-Benchmark. The reason is that AdtIdt only requires video matting results without ground truth.

The comparison of opaque regions (semantics) and mixed regions (details) is presented in Table 1. For MAD and RMSD, in non-data-driven based methods, it can be observed that sampling-based ones such as Robust (single-frame sampling), Robust (single-frame), Robust3D, and Sparse3D generally exhibit poor opaque performance, which aligns with the concept that sampling methods are not suitable for solving classification problems. On the other hand, various affinity-based methods such as CLF3D, KNNLap, and Nonlocal show similar or better performance for opaque pixels. For mixed pixels, the effect of sampling-based methods is similar to that of affinity-based ones. For data-driven based methods, FT-PVM adopts an independent semantic trimap generation branch in the network, which obviously has excellent discriminative ability for opaque pixels. Compared to DVM and TCVOM, CRGNN aggregate more adjacent frames

Table 1
Comparison of various methods within opaque and mixed regions over three evaluation metrics and two trimap sizes, where each data represents the average of all frames from 33 or 41 training video sequences, and bold indicates the minimum value in the current column.

Methods	Opaque regions (Semantics)						Mixed regions (Details)					
	MAD		RMSD		AdtIdt		MAD		RMSD		AdtIdt	
	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small
Sparse3D	4.8	4.47	18.5	17.1	11.0	10.8	14.0	12.7	25.0	22.9	29.8	27.6
Robust (Single-frame)	1.7	1.1	10.6	7.8	4.6	4.0	12.7	9.8	23.9	19.5	22.3	17.8
Robust (Single-frame filter)	1.2	0.8	8.4	6.3	2.2	2.3	12.9	10.6	23.7	20.2	15.8	13.3
Nonlocal	1.8	2.3	9.0	10.1	2.8	3.6	16.7	12.4	26.5	21.4	20.6	16.3
CLF3D	1.5	1.6	8.4	8.6	2.4	1.9	14.5	10.9	24.4	20.2	11.2	8.0
Robust3D	1.3	0.9	9.0	6.5	2.4	2.3	12.5	9.5	23.3	18.7	12.4	10.2
KNNLap	1.3	1.8	8.2	9.4	10.9	8.6	7.7	6.4	18.7	17.1	26.0	18.4
TCVOM	9.7	11.0	21.0	20.5	3.3	4.9	6.9	4.9	17.5	14.2	15.0	13.8
CRGNN	2.6	1.1	11.6	5.3	3.5	2.3	**5.1**	**3.8**	**14.2**	**10.5**	11.9	8.1
FTPVM	**0.6**	**0.6**	4.8	**4.3**	6.2	4.7	9.5	7.7	18.4	15.2	20.2	15.8
1st-our	0.8	1.0	**4.1**	4.8	3.3	3.1	9.1	7.2	18.2	16.0	15.3	12.2
2nd-our	1.5	1.4	9.2	8.2	9.1	6.8	7.9	5.4	18.4	14.5	22.0	14.8
NHVM (3rd-our)	0.9	1.0	5.5	6.0	**2.0**	**1.9**	6.1	4.6	15.0	12.5	**8.6**	**6.9**

Table 2
Comparison of video matting results of various methods over three evaluation metrics and two trimap sizes on two test datasets.

Methods	VideoMatte240K						VM-Benchmark					
	MAD		RMSD		AdtIdt		MAD		RMSD		AdtIdt	
	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small
Sparse3D	17.3	15.5	30.2	27.6	46.7	41.6	23.0	21.9	35.0	33.5	29.2	31.9
Robust (Single-frame)	13.9	10.3	26.0	20.6	30.2	23.7	15.6	12.9	26.9	22.7	21.6	18.8
Robust (Single-frame filter)	13.8	10.9	25.4	21.2	21.3	17.7	14.9	12.5	24.9	21.3	13.0	12.1
CLF3D	15.1	11.7	25.5	21.6	16.4	11.8	18.5	14.7	27.8	24.1	9.3	7.1
Robust3D	13.5	9.8	25.0	19.4	16.0	13.2	14.8	12.2	25.4	21.3	12.9	11.6
KNNLap	8.6	7.9	20.7	20.0	49.2	35.8	10.4	9.0	20.4	18.6	17.7	13.4
Nonlocal	18.2	14.2	28.7	24.5	31.7	26.9	19.2	15.9	28.9	25.0	10.4	9.0
DVM	20.0	13.7	38.3	28.3	25.0	16.2	24.3	17.9	41.5	31.8	20.0	16.9
TCVOM	19.3	18.6	32.1	29.3	24.6	26.6	**8.2**	7.3	**18.4**	16.4	**8.4**	**6.3**
CRGNN	6.7	**4.3**	17.9	**11.1**	15.0	9.7	10.9	**6.5**	22.1	**15.0**	16.0	11.3
FTPVM	9.4	7.9	18.9	15.6	30.3	22.0	12.3	9.6	20.6	17.5	20.4	18.0
NHVM (our)	**6.4**	5.0	**15.4**	13.5	**10.9**	**9.4**	9.0	7.4	18.8	16.0	10.3	7.7

for the current frame, thus possessing better ability to solve mixed pixels. For our method, it is evident that the first preprocessing step performs well in solving the semantic level of opaque pixels, while the second preprocessing step excels in solving mixed pixels, and the entire solution capability of the final result is still relatively good.

For temporal coherence AdtIdt, it is clear that the two-frame Robust3D (i.e., incorporating CLF3D) aimed at inter-frame coherence and Robust filtered in post-processing show better performance than Robust in single-frame, and far superior to the Robust in single-frame sampling version. This aligns with the goal of enhancing visual coherence. It is worth noting that the AdtIdt metric for local affinity-based methods such as CLF3D and Nonlocal is perfect. It is worth noting that data-driven based methods are not significantly different from non-data-driven methods for AdtIdt, and are even slightly inferior in opaque regions, because feature aggregation and attention can enhance the accuracy of the current frame, but cannot provide significant help for inter-frame correlation. Even the help of any loss function can only provide limited improvement. On the contrary, affinity-based methods as well as ours directly adopt affinity correlation and joint solution between adjacent frames, which improves the

temporal inter-frame metric. Finally, the pre-processing layer of our method appears to be slightly worse for the AdtIdt of opaque regions, but the final matting results are still optimal.

The comparison of matting results for the entire frames is presented in Table 2, where region S in Eq. 13 corresponds to the entire frame. For clarity, Table 2 displays the two datasets separately. Note that VideoMatte240K is a portrait dataset with a large number of opaque pixels, while VM-Benchmark is a general object dataset with a large number of unknown and mixed pixels where MAD and RMSD of each method will be weaker. In non-data-driven based methods, the affinity-based CLF3D with a small kernel width achieves better quality in terms of temporal coherence (AdtIdt), and KNNLap based on long-distance search has certain advantages in accuracy (MAD & RMSD). The sampling-based Robust3D post-processed by CLF3D also performs tolerable, even in Robust with single-frame and with filter. Note that the poor sampling strategy in Sparse3D leads to poor matting quality. In data-driven based methods, DVM is limited by the training video dataset and augmentations, and TCVOM clearly performs well but overfits on generalized set VM-Benchmark and cannot accurately handle portrait set VideoMatte240K. CRGNN can still achieve better accuracy by solving correlations between more adjacent frames in a graph neural framework, while FTPVM, although good at accurately solving opaque pixels, has no advantage in solving the overall regions. Our proposed method is clearly the most advantageous for the portrait set VideoMatte240K, and still ranks second in comprehensive metrics on VM-Benchmark.

4.3.2. Ablations

To demonstrate the performance of each layer in the proposed method, some intermediate levels and experimental results for ablation studies are introduced against the final result and shown in Table 3. The ablations for 1st layer (pre-processing) have been presented in the previous section. For the comparative experiments against the 2^{nd} layer (sampling), the following two ablations are introduced. (a) Given that a similar sampling strategy is proposed in our 2nd layer as shared sampling, the original shared matting combined with our 1^{st} layer is also introduced. It is clear that the sampling prior in our 2^{nd} layer significantly improves the final results compared to the shared sampling prior, due to the use of a richer set of optimized sample pairs for mixed pixels, as well as a better opaque pixel scoring and classification mechanism. Additionally, our 2^{nd} layer adopts an inter-frame sampling framework based on optical flow, resulting in better frame coherence. (b) Our 1st layer is also directly combined with the 3^{rd} affinity-based matting solver while ignoring the 2^{nd} layer. The experiments shows that this level is superior to the method with only sampling-based layer, i.e., $1^{st} + 2^{nd}$, but inferior to the combination of sampling- and affinity-based methods (final result). It is clear that the sampling framework in our 2^{nd} layer could have a certain improvement on the final results. Note that it is still weak by using only sampling on temporal coherence AdtIdt.

For our 3^{rd} layer, a method combining our 1^{st} and 2^{nd} layers is employed and integrated with three different search strategies of 3D dual-frame Matting Laplacian for ablation study and comparison. These strategies are: (a) nearest neighbor strategy (i.e. CLF3D), (b) medium-distance nonlocal search strategy (denoted as NonCLF), and (c) long-distance two-stage KNN search strategy (denoted as KNNCLF). It can be observed that these three strategies with fixed search ranges perform unsatisfactorily under all the evaluation metrics, whereas the adaptive search approach proposed in our 3^{rd} layer is capable of handling various foreground boundary scenarios.

4.3.3. Comparison in individual video sequences

Figure 5 shows the comparison on MAD, RMSD and AdtIdt over various video matting methods under two trimap types, where S represents the full-frame set, and the first three and last three columns represent non-data-driven and data-driven methods, respectively. Additionally, the pre-processing layer (layer 1), the pre-processing layer combined with sampling layer (layers 1+2), and the final result (layers 1+2+3) are also shown for comparison. Columns 1-25 on the horizontal axis in each metric correspond to the 25 video sequences in VideoMatte240. Besides, columns 26-33 on the horizontal axis of each MAD and RMSD list the eight training sequences in the VM dataset, and column 26-41 on the horizontal axis of AdtIdt list the above eight training sequences and eight additional test sequences in the VM dataset. Intuitively, these two metrics decrease as the trimap size decreases, indicating an improvement in quality. In terms of accuracy, VideoMatte240 consists of only portraits, leading to relatively smooth changes in both MAD and RMSD. In terms of AdtIdt, the effects of the two datasets are similar. However, due to the complexity of foreground object changes in the VM dataset, in addition to hard foreground boundaries, it also includes soft boundaries such as hair, translucent objects or motion blur, resulting in relatively drastic changes in MAD and RMSD. As discussed, compared to other methods, the performance of our method is relatively excellent at the non-data-driven level, and it will gradually improve from layer 1 to layer 3. For data-driven level, the proposed method is still competitive.

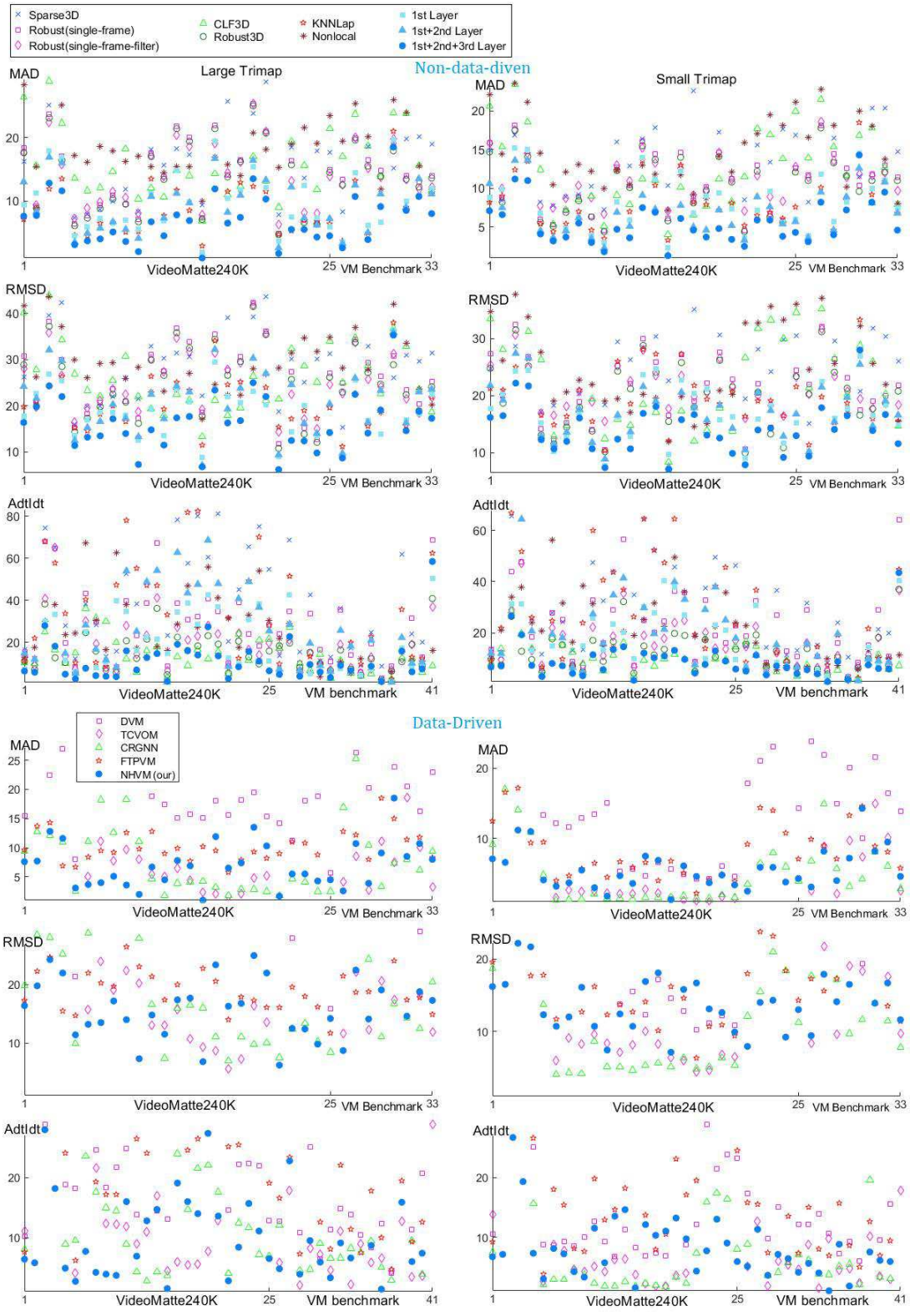


Figure 5: Comparison on the average MAD, RMSD and AdtIdt of video matting results obtained from non-data-driven based (former 3 rows) and data-driven based (latter 3 rows) methods within two trimap sizes, across sequences of VideoMatte240 and VM-Benchmark datasets.

Table 3
Comparison of ablation tests of our methods over three evaluation metrics and two trimap sizes on two test datasets.

Methods	VideoMatte240K						VM-Benchmark					
	MAD		RMSD		AdtIdt		MAD		RMSD		AdtIdt	
	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small	Large	Small
1st layer (pre-processing)	9.7	7.8	18.6	16.6	23.1	18.7	10.4	9.1	20.0	17.9	11.4	9.9
1st layer + Shared	9.8	7.5	38.3	28.3	25.0	16.2	11.1	8.9	41.5	31.8	20.0	16.9
1st layer + 2nd layer	9.0	6.4	20.6	16.6	36.6	24.7	10.5	8.1	21.8	17.8	22.4	16.6
1st layer + 3rd layer	7.4	5.9	32.1	29.3	24.6	26.6	9.8	8.1	18.4	16.4	**8.4**	**6.3**
1st layer + 2nd layer + NonCLF	12.5	10.9	17.9	**11.1**	15.0	9.7	11.9	10.2	22.1	15.0	16.0	11.3
1st layer + 2nd layer + KNNCLF	13.5	12.8	14.2	11.5	20.0	13.0	11.7	10.5	**17.6**	**12.1**	9.3	7.3
1st layer + 2nd layer + CLF3D	8.9	7.3	18.9	15.6	30.3	22.0	11.9	10.2	20.6	17.5	20.4	18.0
1st layer + 2nd layer+ 3rd layer (our)	**6.4**	**5.0**	**15.4**	13.5	**10.9**	**9.4**	**9.0**	**7.4**	18.8	16.0	10.3	7.7

4.4. Qualitative comparison

In qualitative comparison of visual perception, the results of the newly introduced single-frame methods in the ablation experiment are excluded and the remaining existing dual-frame video matting methods, as well as our 1st layer and the final results. Figure 6, Figure 7, Figure 8 and Figure 9 show several examples among these methods. In the first row of the input images, blue and red lines denote the known foreground and background edges of the input trimaps.

For non-data-driven based methods, Figure 6 and Figure 7 illustrates 5 examples extracted from a certain frame in 5 video sequences from VM-Benchmark and VideoMatte240 datasets respectively. Sampling-based methods such as Sparse3D and Robust are limited by their imprecise sampling strategies, resulting in a significant number of noises in the opaque results and far less ideal visual performance. Robust3D, on the other hand, benefits from the matting equation and has shown promising results in solving translucent pixels. Affinity-based methods such as CLF3D, Nonlocal are still limited by their respective locally biased mapping benchmarks, resulting in unsatisfactory results for opaque regions and large pieces of blurring in large translucent regions.

Correspondingly, KNNLap and our 1st layer on preprocessing both utilize long-range search to discover samples with similar colors, which can more accurately locate the linear relationships between pixels. Nevertheless, there is still a lack of precision in transparency. Our final results benefit not only from multi-level searches but also from the precise sampling priors and multi-level affinities, thus achieving the best results in the precision of both opaque and mixed pixels and visual effects.

For data-driven based methods, Figure 7 and Figure 8 illustrates 5 examples extracted from a certain frame in 5 video sequences from VM-Benchmark and VideoMatte240 datasets respectively. Generally, for VM-Benchmark, due to the semantic limitations of the training set and the neglect of matting equation, various deep-based methods include certain errors in terms of semantics and details, such as excessive similarity between fore- and background, or overly wide translucent areas. For VideoMatte240, due to the widespread applications of portrait-based training sets and the frequently occurrence of hard boundaries, the semantic results are generally good among various methods. Despite some flaws, the overall performance of our approach on both datasets can still compete with data-driven methods.

For practical application, the corresponding methods require the simultaneous calculation of the two components $\{\alpha, \boldsymbol{F}\}$ for each frame to be applied to the final synthesis. Among the aforementioned methods, only Robust3D can solve the optimal foreground and background values corresponding to the intermediate optimal sample pairs from the sampling step, while other methods are unable to obtain foreground. As mentioned, our sampling step can also record the optimal foreground and background samples for the current pixel. Therefore, only these two methods can be compared for the final synthesis effect. To this end, four representative video sequences from the video matting dataset are selected and their $\{\alpha, \boldsymbol{F}\}$ of each frame are computed for each method.

Additionally, four high resolution backgrounds were selected from the BG-20K dataset to serve as new backgrounds for the synthesized images. Figure 10 shows the synthetic frames of Robust3D and our method on four sets of video frames. Clearly, the visual perception of the synthesized frames by our method is significantly better than that of Robust3D. Firstly, the α results of our method are more precise, manifested in the absence of many noises in opaque pixels and a more refined solution for transparency. Secondly, the foreground results of our method are also precise, as demonstrated in the translucent regions of the second and third of video sets.

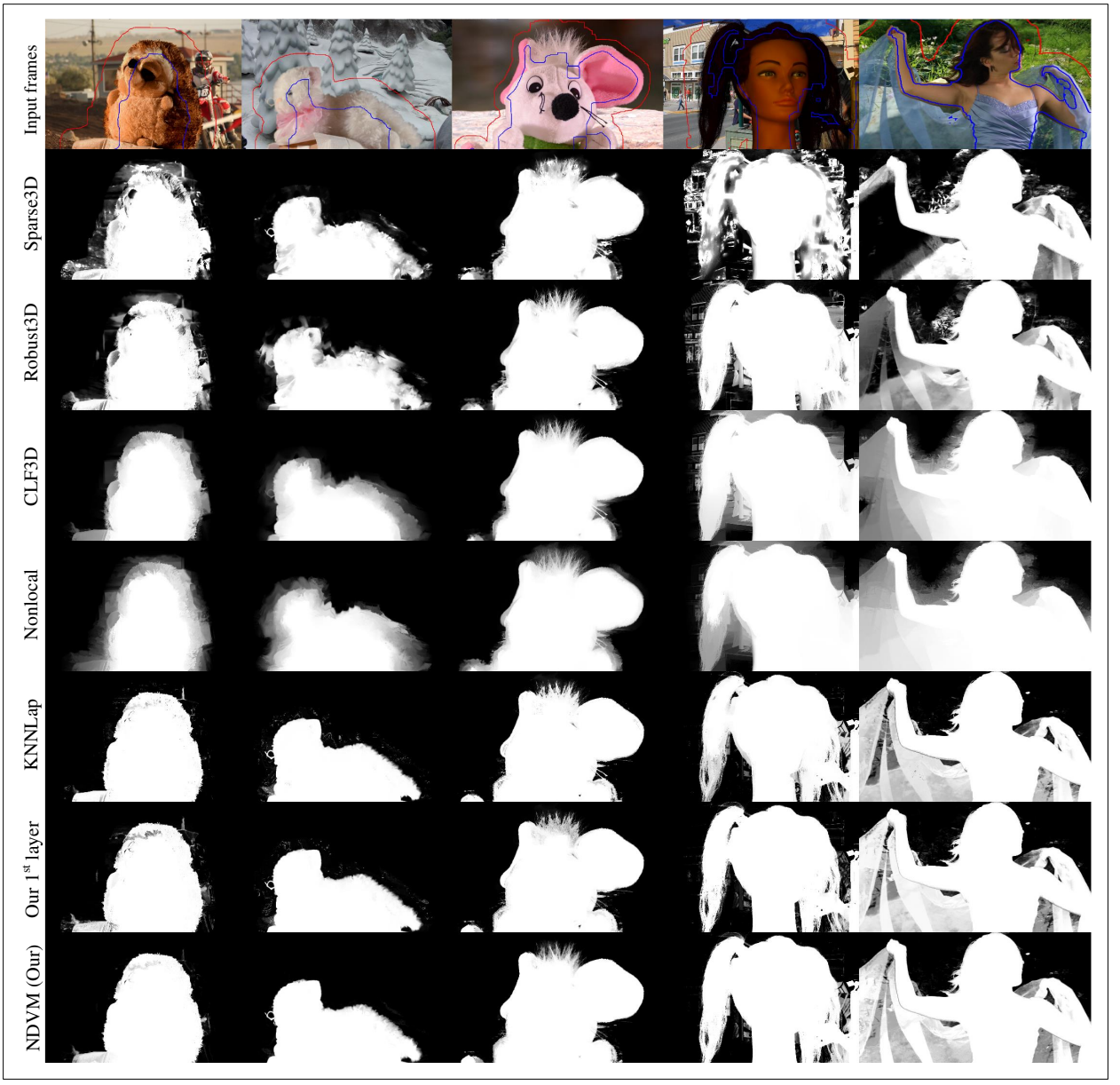


Figure 6: Comparisons of 5 frames of video matting results for various non-data-driven based video matting methods in the sequences of VM-Benchmark dataset.

5. Conclusion

Currently, data-driven methods have shown certain advantages over non-data-driven methods in video matting, but they also present certain problems such as ignoring the matting equation and imprecise temporal coherence. This paper fully utilizes the advantages of non-data-driven based approaches and proposes a non-data-driven hierarchical framework on video matting, achieving high matting quality in both intra-frame accuracy and inter-frame coherence. The fundamental concept is the different processing approaches adopted for opaque and mixed pixels. The first layer is an affinity-based pre-processing method for consecutive frames, which can provide more accurate opaque samples for subsequent steps. The second layer combines two-frame sampling with optical flow, leveraging the precise propagation in opaque pixels to provide priors for both opaque and mixed pixels. The third layer employs a dual-frame affinity-based method that utilizes the preprocessed trimap shape, selects the most suitable search for adaptive settings of kernel width,

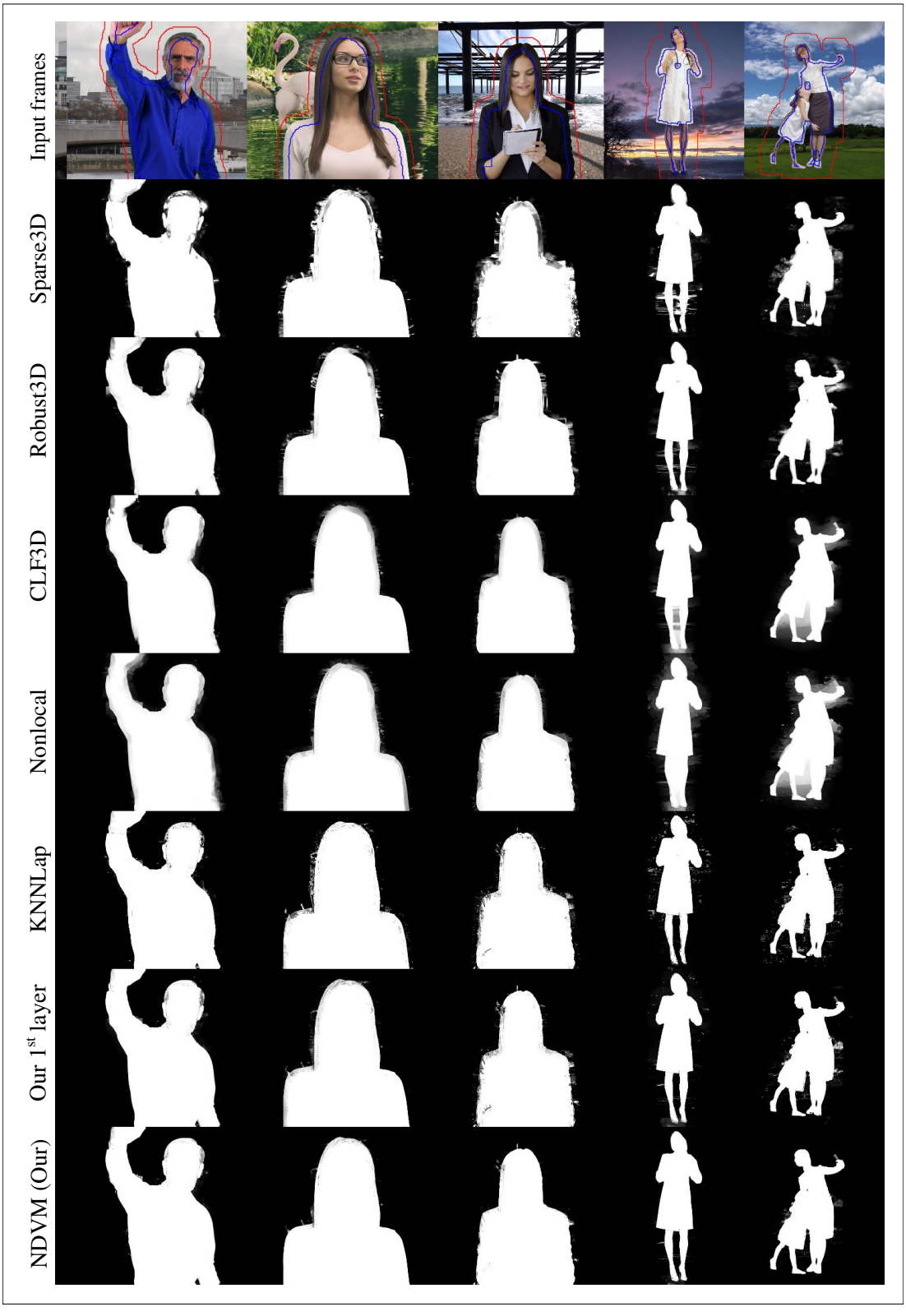


Figure 7: Comparisons of 5 frames of video matting results for various non-data-driven based video matting methods in the sequences of VideoMatte240K dataset.

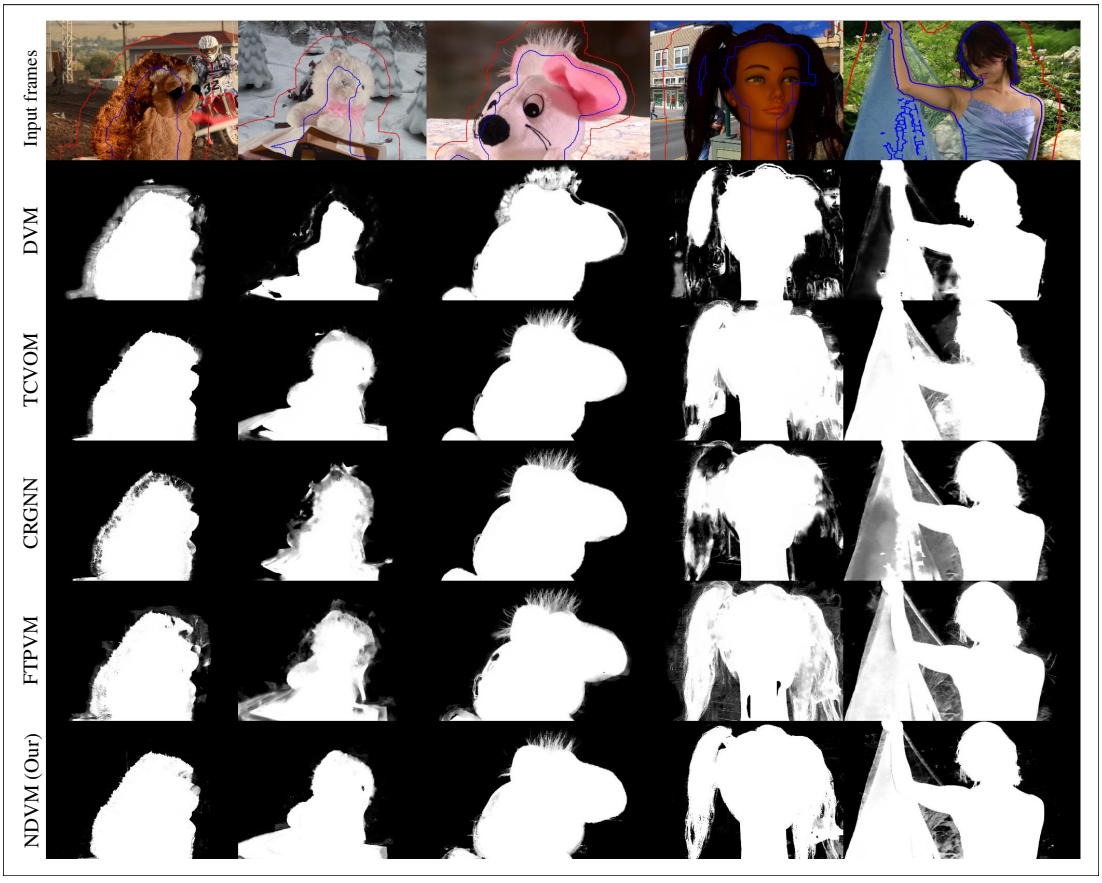


Figure 8: Comparisons of 5 frames of video matting results for data-driven based video matting methods in the sequences of VM-Benchmark dataset.

and ultimately solves the video matting problem. Experimental results and ablation studies demonstrate that the final output of this method exhibits significant advantages over existing methods, both quantitatively and qualitatively. Additionally, this method can also be adapted to synthetic applications of virtual reality in the context of the metaverse.

6. Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. Acknowledgments

This work is supported by Natural Science Foundation of Heilongjiang Province (No. PL2024F004).

References

Aksoy, Y., Aydun, T.O., Pollefeys, M., 2017. Designing effective inter-pixel information flow for natural image matting, in: IEEE Conference on Computer Vision and Pattern Recognition.

Backes, M., Oliveira, M., 2019. A patchmatch-based approach for matte propagation in videos. Pacific Graphics 38, 651–662.

Bai, X., Sapiro, G., 2009. Geodesic matting: A framework for fast interactive image and video segmentation and matting. International Journal of Computer Vision 82, 113–132.

Bai, X., Wang, J., Simons, D., 2011. Towards temporally coherent video matting. Computer Vision/Computer Graphics Collaboration Techniques , 63–74.

Bai, X., Wang, J., Simons, D., Sapiro, G., 2009. Video snapcut: Robust video object cutout using localized classifiers. ACM Transactions on Graphics 28, 1–11.

Cai, S., Zhang, X., Fan, H., Huang, H., Liu, J., Liu, J., Liu, J., Wang, J., Sun, J., 2019. Disentangled image matting, in: IEEE International Conference on Computer Vision, pp. 8819–8828.

Chen, Q., Ge, T., Xu, Y., Zhang, Z., Yang, X., Gai, K., 2018. Semantic human matting, in: 26th ACM Multimedia conference, MM 2018, pp. 618–626.

Chen, Q., Li, D., Tang, C., 2013. KNN matting. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 2175–2188.

Choi, I., Lee, M., Tai, Y., 2012. Video matting using multi-frame nonlocal matting laplacian, in: European Conference on Computer Vision, pp. 540–553.

Chuang, Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R., 2002. Video matting of complex scenes. ACM Transactions on Graphics



Figure 9: Comparisons of 5 frames of video matting results for data-driven based video matting methods in the sequences of VideoMatte240K dataset.

21, 243–248.

Chuang, Y., Curless, B., Salesin, D., Szeliski, R., 2001. Geodesic matting: a framework for fast interactive image and video segmentation and matting, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 264–271.

Erofeev, M., Gitman, Y., Vatolin, D., Fedorov, A., Wang, J., 2015. Perceptually motivated benchmark for video matting, in: British Machine Vision Conference, pp. 1–15.

Gastal, E., Oliveira, M., 2010. Shared sampling for real-time alpha matting. Computer Graphics Forum 29, 575–584.

He, K., Rhemann, C., Rother, C., Tang, X., Sun, J., 2011. A global sampling method for alpha matting, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2049–2056.

Hou, Q., Wang, C., 2021. Automatic portrait video matting via context motion network. arXiv:arXiv 2109.04598.

Hu, Y., Lin, Y., Wang, W., Zhao, Y., Wei, Y., Shi, H., 2024. Diffusion for natural image matting, in: European Conference on Computer Vision, pp. 181–199.

Huang, W., Lee, M., 2023. End-to-end video matting with trimap propagation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Johnson, J., Rajan, D., Cholakkal, H., 2015. Temporal trimap propagation using motion-assisted shape blending, in: Visual Communications and Image Processing (VCIP), pp. 1–4.



Figure 10: Comparison of 4 video synthesis results between our method and Robust3D on 4 video frames.

Karacan, L., Erdem, A., Erdem, E., 2017. Alpha matting with kl-divergence based sparse sampling. IEEE Transactions on Image Processing 26, 4523–4536.

Ke, Z., Sun, J., Li, K., 2022. Modnet: Real-time trimap-free portrait matting via objective decomposition, in: AAAI Conference on Artificial Intelligence, pp. 1140–1147.

Lee, S., Yoon, J., Lee, I., 2010. Temporally coherent video matting. Graphical Models 72, 1–19.

Levin, A., Lischinski, D., Weiss, Y., 2008. A closed-form solution to natural image matting. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 228–242.

Li, D., Chen, Q., Tang, C., 2013. Motion-aware knn laplacian for video matting, in: IEEE International Conference on Computer Vision, pp. 1–8.

Li, J., Goel, V., Ohanyan, M., Navasardyan, S., Wei, Y., Shi, H., 2024. Vmformer: end-to-end video matting with transformer, in: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 6664–6673.

Li, J., Ohanyan, M., Goel, V., Navasardyan, S., Wei, Y., Shi, H., 2023. Videomatt: A simple baseline for accessible real-time video matting, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2177–2186.

Li, J., Zhang, J., Maybank, S., Tao, D., 2022. Bridging composite and real: Towards end-to-end deep image matting. International Journal of Computer Vision 130, 246–266.

Li, Y., Lu, H., 2020. Natural image matting via guided contextual attention, in: AAAI Conference on Artificial Intelligence, pp. 11450–11457.

Li, Y., Sun, J., Shum, H., 2005. Video object cut and paste. ACM Transactions on Graphics 24, 595–600.

Lin, S., Ryabtsev, A., Sengupta, S., 2021. Real-time high-resolution background matting, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8762–8771.

Lin, S., Yang, L., Saleemi, I., 2022. Robust high-resolution video matting with temporal guidance, in: IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 238–247.

Liu, J., Yao, Y., Hou, W., 2020. Boosting semantic human matting with coarse annotations, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8563–8572.

Lu, H., Dai, Y., Shen, C., Xu, S., 2019. Indices matter: Learning to index for deep image matting, in: IEEE International Conference on Computer Vision, pp. 3266–3275.

Park, G., Son, S., Yoo, J., 2022. Matteformer: Transformer-based image matting via prior-tokens, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11696–11706.

Sengupta, S., Jayaram, V., Curless, B., Seitz, S., Kemelmacher-Shlizerman, I., 2020. Background matting: The world is your green screen, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–16.

Seong, H., Oh, S.W., Price, B., Kim, E., Lee, J.Y., 2022. One-trimap video matting, in: 17th European Conference on Computer Vision, pp. 430–448.

Shahrian, E., Price, B., Cohen, S., Rajan, D., 2012. Temporally coherent and spatially accurate video matting. Computer Graphics Forum 33, 381–390.

Shen, X., Tao, X., Gao, H., Zhou, C., Jia, J., 2016. Deep automatic portrait matting, in: European Conference on Computer Vision, pp. 92–107.

Sindeev, M., Konushin, A., Rother, C., 2012. Alpha-flow for video matting, in: the 11th Asian Conference on Computer Vision, pp. 438–452.

Sun, Y., Tang, C.K., Tai, Y.W., 2023. Ultrahigh resolution image/video matting with spatio-temporal sparsity, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14112–14121.

Sun, Y., Wang, G., Gu, Q., 2021. Deep video matting via spatio-temporal alignment and aggregation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6975–6984.

Tang, Z., Miao, Z., Wan, Y., Zhang, D., 2012. Video matting via opacity propagation. The Visual Computer 28, 47–61.

Wang, J., Bhat, P., Colburn, R., Agrawala, M., Cohen, M., 2005. Interactive video cutout. acm transactions on graphics. Image and Vision Computing 24, 585–594.

Wang, J., Cohen, M., 2007a. Image and video matting: A survey. Foundations and Trends in Computer Graphics and Vision 3, 97–175.

Wang, J., Cohen, M., 2007b. Optimized color sampling for robust matting, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Wang, T., Liu, S., Tian, Y., 2021. Video matting via consistency-regularized graph neural networks, in: IEEE/CVF International Conference on Computer Vision, pp. 4902–4911.

Xu, N., Price, B., Cohen, S., 2017. Deep image matting, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Yao, J., Wang, X., Yang, S., Wang, B., 2024a. Vitmatte: Boosting image matting with pre-trained plain vision transformers. Information Fusion 103, 13.

Yao, J., Wang, X., Ye, L., Liu, W., 2024b. Matte anything: Interactive natural image matting with segment anything models. Image and Vision Computing 147.

Zhang, Y., Wang, C., Cui, M., 2021. Attention-guided temporally coherent video object matting, in: 29th ACM International Conference on Multimedia, pp. 5128–5137.

Zou, D., Chen, X., Cao, G., Wang, X., 2020. Unsupervised video matting via sparse and low-rank representation. IEEE Transactions on Pattern and Machine Intelligence 42, 1501–1514.