FactorMatte: Redefining Video Matting for Re-Composition Tasks

ZEQI GU, Cornell Tech, USA
WENQI XIAN, Cornell Tech, USA
NOAH SNAVELY, Cornell Tech, USA
ABE DAVIS, Cornell University, USA

Fig. 1. We show that by re-framing the matting problem as counterfactual video synthesis we can produce useful decompositions of video with complex cross-component interactions. Here, the input video (left) shows a child running through a puddle to create splashes. Our method factors the video into two counterfactual components: one for the foot in the foreground (single layer, top left) and another for the background, which includes static parts of the environment (top middle) as well as the splash (top right). The bottom row shows three re-combinations of the component layers, and the image on the right shows a re-composition where the foreground layer has been replaced with a virtual object. To our knowledge, no previous matting method can handle these kinds of complex cross-component interactions. Video credits to Youtube user @trucksandplay8446.

We propose Factor Matting, an alternative formulation of the video matting problem in terms of counterfactual video synthesis that is better suited for re-composition tasks. The goal of factor matting is to separate the contents of a video into independent components, each representing a counterfactual version of the scene where the contents of other components have been removed. We show that factor matting maps well to a more general Bayesian framing of the matting problem that accounts for complex conditional interactions between layers. Based on this observation, we present a method for solving the factor matting problem that learns augmented patch-based appearance priors to produce useful decompositions even for video with complex cross-layer interactions like splashes, shadows, and reflections. Our method is trained per-video and does not require external training data or any knowledge about the 3D structure of the scene. Through extensive experiments, we show that it is able to produce useful decompositions of scenes with such complex interactions while performing competitively on classical matting tasks as well. We also demonstrate the benefits of our approach on a wide range of downstream video editing tasks. Our project website is at: https://factormatte.github.io/.

CCS Concepts: • Computing methodologies → Computer vision representations: Rendering; Neural networks.

Additional Key Words and Phrases: matting, video matting, compositing, video layer decomposition

ACM Reference Format:
Zeqi Gu, Wenqi Xian, Noah Snavely, and Abe Davis. 2023. FactorMatte: Redefining Video Matting for Re-Composition Tasks. ACM Trans. Graph. 42, 4, Article 47 (August 2023), 14 pages. https://doi.org/10.1145/3592423

1 INTRODUCTION

The origins of video matting date back to the late 1800’s—long before the invention of digital video—with the term "matting" itself first referring to the application of matte paint to glass plates that were set in front of a camera during filming. Once painted, the matte portion of a plate would block parts of the frame from being exposed. The filmmaker could then expose the same film multiple times, using different plates to fill in different parts of the scene with different exposures. More than a century later, modern digital matting functions much the same way, digitally mixing images according to the compositing equation:

\[ v_p = \alpha_p F_p + (1 - \alpha_p) B_p \]  \hspace{1cm} (1)

which defines an output image value \( v_p \) at image location \( p \) as a linear combination of a foreground color \( F_p \) and a background color \( B_p \). This equation closely mimics the behavior of its analog predecessor, with \( \alpha_p \) representing the transparency of a glass plate used to film the foreground, and \( (1 - \alpha_p) \) representing the transparency of a plate used to film the background. Compared to glass plates, digital mattes are simple to manipulate and need not remain static over...
time, which makes it easy to re-compose scenes by transforming and re-combining matted content from different sources. This type of re-compositing can be very powerful under the right conditions, but has some significant limitations as well. Equation 1 assumes that the content of a scene can be factored unambiguously into foreground and background layers that are independent, which is problematic in cases where these layers interact with each other in complex ways—for example, in the scenario shown in Fig. 1 where a foot (foreground) interacts with water (background) to create a splash, the foreground and background are clearly not independent.

In this paper, we propose Factor Matting as an adjusted framing of the matting problem. The goal is to factor input video into distinct components, each representing a counterfactual version of the scene with different content removed. We relate each counterfactual component to a conditional prior on the appearance of different scene content. We show that factor matting closely follows a Bayesian formulation of the matting problem where common limiting assumptions about the independence of different layers have been removed. We then present FactorMatte, a solution to the factor matting problem that offers a convenient framework for generalizing classical matting priors to conditional ones based on expected deformations in a scene. We show that even without training on external data, FactorMatte yields competitive results on traditional matting problems and produces useful decompositions for inputs with complex foreground-background interactions that have not been addressed in previous works.

2 RELATED WORK

2.1 Matting Using Priors

The traditional formulation of matting involves inverting Equation 1 to recover per-pixel alpha values. As this problem is under-constrained, solving it requires some sort of prior; it can be useful to interpret previous approaches in terms of the priors they employ.

2.1.1 Color-Based Priors. Priors based solely on color have the advantage of being applicable to static image matting. One of the earliest approaches of this type is color keying, which adopts the prior that the background will be a particular key color [Smith and Blinn 1996]. This strategy is effective when one can control the background, for example by filming in front of a green screen. Hence, color keying is still commonly used by filmmakers today.

Scenarios where the background cannot be controlled are often referred to as natural image matting. Ruzon and Tomasi were among the first to tackle this problem [Ruzon and Tomasi 2000]. Their approach asks a user to partition images into three regions: unambiguous foreground pixels, unambiguous background pixels, and pixels that are an unknown mixture of these two layers. This segmentation, often referred to as a trimap, provides a labeled set of representative samples from the foreground and background with which to train data-driven priors. Chuang et al. build on this approach by introducing a general Bayesian framework for such problems [Chuang et al. 2001]. Under this framework, matting is performed by solving for a maximum a posteriori (MAP) estimate of the alpha value at each pixel in the uncertain region of the provided trimap. While our method is quite different from these early approaches, we motivate our approach by their Bayesian reasoning (Sec. 3).

Later image matting approaches focused on operating with more relaxed input labels [Wang and Cohen 2005], or optimizing according to detail representations such as image gradients [Sun et al. 2004] or image Laplacians [Levin et al. 2008].

2.1.2 Motion-Based Priors. The simplest motion-based prior is basic background subtraction, which assumes that the background content in a scene does not change over time [Barnich and Van Droogenbroeck 2010; Qian and Sezan 1999]. Other early video matting methods use optical flow to interpolate and propagate trimaps across different frames of video (e.g. [Chuang et al. 2002]). Subsequent works make more explicit use of temporal coherence by looking at video matting as an operation over a space-time volume of pixels [Bai et al. 2011, 2009; Choi et al. 2012; Lee et al. 2010; Li et al. 2013]. More recent methods have employed background subtraction within neural network frameworks [Lim and Keles 2018, 2020; Tezcan et al. 2020]. Other recent background matting methods achieve real-time, high-resolution matting but still require a clean background reference image [Lin et al. 2021; Sengupta et al. 2020]. Most recently, Omnimatte uses a variant of background subtraction that initially fits the video to a static background transformed with per-frame homographies [Lu et al. 2021]. This variant of background subtraction is much more robust to camera rotation, but still struggles with camera translation or motion of background elements.

2.2 Neural Matting

Recent efforts in matting have focused on improving trimap-based results with the power of neural networks [Cho et al. 2016; Forte and Pitié 2020; Hou and Liu 2019; Li and Lu 2020; Lu et al. 2019; Tang et al. 2019; Xu et al. 2017], and extending methods to video [Lin et al. 2022; Sun et al. 2021]. The training data for video matting requires precise labeling of details, such as strands of hair, making it difficult to acquire at scale. Therefore, many methods train on synthetic data where ground-truth labels are easy to obtain [Ke et al. 2022; Li and Lu 2020; Lu et al. 2019; Seong et al. 2022; Xu et al. 2017; Zhang et al. 2021]. In contrast, our approach focuses on optimizing for a single input video without training on large external datasets.

2.3 Layered Video Representations

Layered representations of video are useful for many editing tasks. Layered Neural Atlases [Kasten et al. 2021], Marionette [Smirnov et al. 2021] and Deformable Sprites [Ye et al. 2022] decompose videos into multiple layers that deform according to transformations that model non-rigid motion and self-occlusion. However, these methods are limited to objects whose appearance in a video can be mapped to a planar mosaic, so they cannot explicitly handle changes in appearance over time, for instance, due to lighting.

The work most closely related to ours is Omnimatte [Lu et al. 2021], which uses a neural network to decompose a video into a set of color and opacity layers. Omnimatte works by first solving for a background layer that approximates the video as a series of homographies applied to a static image. Then it attributes content to each foreground layer based on correlated motion (e.g., the pixels of a person’s shadow are attributed to the layer initialized with a
mask containing that person). Our work builds on this approach in two key ways. First, we explicitly model the appearance of each layer, which lets us factor pixels that are a mixture of colors from different components into their respective contributions from each layer. Without such a model, Omnimatte often incentivizes the duplication of scene content across multiple layers to minimize reconstruction error (see Fig. 6), which can be useful in re-timing applications but limiting when recomposing with content from a different scene.

Second, and also distinguishing our work from all other matting approaches we are aware of, we relax the implicit assumption that the foreground and background are independent in order to tackle videos with more complex foreground-background interactions.

3 LIMITS OF CLASSICAL MATTING

To motivate our re-framing of the matting problem in Sec. 4, we first review the classic Bayesian formulation from Chuang et al. [2001]

3.1 Bayesian Formulation

Matting is traditionally posed as inference of the spatially-varying parameter \( a_p \) from Equation 1 over a provided image or video. To cover videos, we consider a tensor form of Equation 1:

\[
V = A \otimes F + (1 - A) \otimes B
\]

where \( V \) is a given video, and \( A, F, \) and \( B \) are alpha, foreground, and background tensors with dimensions matching \( V \). We can formulate the solution to our matting problem as a MAP (maximum a posteriori) estimation of our parameters:

\[
\arg\max_{F, B, A} P(F, B, A|V)
\]

Applying Bayes rule, we get:

\[
\arg\max_{F, B, A} \frac{P(V|F, B, A) P(F, B, A)}{P(V)}
\]

where the likelihood term \( P(V|F, B, A) \) measures whether the contents of \( V, F, B, \) and \( A \) satisfy Equation 2, and \( P(F, B, A) \) is our joint prior over these values.

3.2 Common Assumptions of Classical Matting

Equation 5 can be simplified by setting \( P(V) = 1 \), since our video is given. The classical formulation further assumes a uniform prior on alpha values independent of foreground and background color, and subsequent approaches have either maintained this assumption or implicitly incorporated joint information about the alpha channel into foreground and background priors trained on external data. These observations then simplify Equation 4 to

\[
\arg\max_{F, B, A} \frac{P(V|F, B, A) P(F, B)}{P(F, B)}
\]

Previous matting approaches have also assumed solutions that do not satisfy Equation 2 have a very low likelihood. As the values \( v_p \in V \) are observed, this effectively restricts the foreground and background values for each pixel, \( \{F_p \in F, B_p \in B\} \) to pairs that contain \( v_p \in V \) in their convex span (defined by adherence to the per-pixel Equation 1). Even with this restriction, the matting problem remains under-constrained, yielding many possible solutions. This ambiguity can be resolved by maximizing the prior \( P(F, B) \) over likely solutions. Learning a joint prior \( P(F, B) \) directly from \( V \) is usually impractical due to sample complexity, so most matting approaches assume \( F \) and \( B \) are independent, allowing the separation of their joint distribution into a product of marginal priors:

\[
\arg\max_{F, B, A} P(V|F, B, A) P(F) P(B)
\]

This last simplification aligns with the assumption that we can transform each matted component in Equation 1 independently. In most applications, marginal priors for each component are derived from either a representative sampling of pixel labels (e.g., a trimap or labeled strokes) or assumptions about the motion of different layers (e.g., background subtraction, where the background is motionless). Unfortunately, both of these strategies fail to account for visible interactions between components, which tend to yield pixels with ambiguous, non-binary labels. We can understand these limits as a failure of marginal priors to capture effects that are only likely when one component is conditioned on an interaction with another.

3.3 Conditional Interactions

The use of marginal priors in Equation 6 assumes independence of the form:

\[
P(B|F) = P(B), \quad P(F|B) = P(F)
\]

Here, \( P(B|F) = P(B) \) implies that the presence of foreground content should not change the appearance of our background. Or equivalently, that visible parts of the background should match that of a counterfactual video where foreground is removed. However, many common effects violate this rule. Take Fig. 2a as an example: here, the background cushion is physically deformed by the foreground cube, and thus its deformed state is improbable without an explanation of the cube causing it.

4 THE FACTOR MATTING PROBLEM

We propose Factor Matte, an adjusted framing of the matting problem that leads to video factorizations well-suited for re-compositing.

Our goal is to factor a video into different components, each showing a counterfactual version of the scene with all but its associated contents removed. For symmetry with previous equations, we formulate this using Bayesian inference and describe the case of factoring a foreground component from the rest of the scene. Other components, including the background, follow symmetrically.

Building from Equation 5, we forego the usual assumption that \( F \) and \( B \) are independent, and instead use the chain rule to factor our joint distribution into the product of a marginal and conditional distribution:

\[
P(F, B) = P(F|B) P(B) = P(B|F) P(F)
\]
Applying this to Equation 5, we get:

$$\argmax_{F, B, A} P(V | F, B, A) \cdot P(F | B) P(B)$$  \hspace{1cm} (9)$$

Here we call the conditional term $P(B | F)$ and its symmetric counterparts $P(F | B)$ conditional priors. Like its marginal counterpart, $P(F | B)$ describes a distribution over the appearance of our foreground. Importantly, while the value of $B$ impacts the distribution that $P(F | B)$ describes, it is not a parameter of that distribution, so $P(F | B)$ is the same for any counterfactual version of our scene that places $F$ in a similar state. Our conditional priors are then analogous to describing the appearance of each component frozen in time with all other components rendered invisible.

To see this logic at play in a background, consider the scene in Fig. 1, which shows a child (foreground) splashing through a puddle (background). Classical matting struggles with this example because a splash would be improbable under marginal priors for the background—spashes do not usually appear without something there to create them. However, such a splash would be considerably more probable in a distribution describing the puddle conditioned on its observed interaction with the foot. Such a distribution would be equivalent to one describing the splash frozen in time with the foot removed. We can check this interpretation by treating the frozen state as an entirely new scene with background $B'$. In this case $P(B') = P(B | F)$, so $P(B' | F) = P(B')$, and we can think of our conditional prior as a marginal prior over a scene where our now-frozen background is really independent of the foreground.

In real scenes, the contribution of a component to video may be sparse in image space; i.e., the ideal alpha channel for that component will have a large number of zeros. This means little or no supervision for the color of that component across much of the video. In practice this biases optimization toward solutions that make other criteria easier to satisfy, typically resulting in correlated color channels across different components (for an example, see Fig. 6). This is problematic if our ultimate goal is to use factored components for downstream re-compositing applications. We can address this by biasing the color of these unsupervised pixels toward values they are likely to take in other compositions. This bias may be application- or scene-dependent and could include things like dark values to represent possible shadows a component might cast in other scenes, or colors representing possible color cast, reflections, illumination, etc.

### 4.1 Comparison to Classical Matting

It can be informative to consider an alternative factorization of our prior into marginal and conditional distributions that is more symmetric but otherwise equivalent to Equation 9:

$$\argmax_{F, B, A} P(V | F, B, A) \cdot [P(F) P(B) P(F | B) P(B | F)]^{\frac{1}{2}}$$  \hspace{1cm} (10)$$

Fig. 2. **Limits of Classical Matting.** The examples here come from a video of a simulated blue cube bouncing off of a yellow cushion. (a) is an input frame when the cube is in contact with and deforming the cushion. (f) shows a static cushion in a rest state for comparison. The remainder of the figure shows a decomposition of this frame into foreground and background content using different methods. (b) and (g) show our results. (c) and (h) show the results of Omnimatte, the video decomposition method closest to ours [Lu et al. 2021]. Omnimatte successfully associates the cube’s shadow with the foreground because the shadow’s apparent motion follows that of the cube, but it also associates the deformation of the background cushion to the foreground for the same reason, thereby leaving this deformation out of the background and instead placing significant background content into the foreground layer. (d) and (i) show the results of a classical image matting method [Chen et al. 2013], which associates shadow pixels with the background as their colors are more similar. Since such matting methods focus on the foreground alpha matte, the background content behind opaque foreground pixels is undefined. (e) and (j) show our results.
If we take the log likelihood of this form we get:

$$\arg\max_{F,B,A} \frac{\mathcal{L}(F) + \mathcal{L}(B) + \mathcal{L}(F|B) + \mathcal{L}(B|F)}{2}$$ (11)

which shows more clearly the relationship to classical matting. The relative weights of each term may vary for different inputs and applications, based on our confidence in the supplied priors.

5 FACTORMATTE

We now describe FactorMatte, our solution to the factor matting problem. Say we want to decompose a video $V$ with $T$ frames into two components: the foreground and the background, then our input is that video along with a rough per-frame segmentation mask indicating what content should be associated with the foreground component (no mask required for the background). Our output is an ordered set of $N$ layers ($L_1, ..., L_N$), where each layer belongs to either the foreground or the background component. Each layer is associated with its own priors and has a unique position in the composition order. This allows content from different components to interleave (e.g., the background component in Fig. 1 has a layer behind the foot and another in front of the foot to represent parts of the splash that occlude the foot). One layer $L_j = (C_j, A_j)$ consists of one color frame at each time, $C_j = (C_j(1), ..., C_j(T))$ and a alpha channel $A_j = (A_j(1), ..., A_j(T))$. For symmetry with previous methods, we focus on describing the base case of factoring videos with one foreground object, and thus one foreground layer for the foreground component. Then we would have two layers for the background component: an environment layer representing static parts of the background, and a residual layer representing all other, more irregular, aspects of the background. This gives us a total of $N = 3$ layers. Our method is designed to accommodate more layers, but we leave this exploration to future work.

Our optimization (Fig. 3) pipeline consists of two modules: a decomposition network $N_D$, and a set of patch-based discriminators $\{D_1, ..., D_q\}$ that represent conditional priors on each layer. It is automatic by default, with optional manual adjustments that can further improve the results. For clarity, we divide our explanation (Sec. 5.1-5.4) into 3 stages: In Stage 1, we use marginal priors to obtain representative samples for each component. In Stage 2, we augment these samples to approximate the conditional effects of other components. Finally in Stage 3, we adversarially train discriminators along with the decomposition network $N_D$ to learn distributions over the augmented samples, leading to a more independent set of factors derived from the input video. In Sec. 5.5, we...
describe some simple additional input a user can give to address especially challenging videos outside the scope of previous work.

5.1 Marginal Priors
The marginal prior for each layer is defined by the pixels we augment to train its conditional prior (discriminator) later. For most layers, this simply consists of the pixels covered by that corresponding input mask. The main exception to this is the environment layer of the background, which we assume can be described by a static image mosaic transformed by a different homography for each frame. To solve for this marginal background in Stage 1, we train the decomposition network \( N_D \) with all losses except those related to the discriminators. The resulting environment color is used as the marginal prior for our background component (i.e. both the environment and the residual layer).

5.2 Conditional Priors
The conditional prior for each layer is represented with a different discriminator, which we train on augmented samples of our marginal prior pixels. These discriminators operate on some feature space, and we must balance the complexity of that feature space against the limited sampling of our marginal priors and augmentations. Here it is illustrative to consider some extreme strategies. Learning a prior over the entire image would result in a very high-dimensional feature space with comparatively few training samples, making it hard to generalize our appearance model to unseen parts of an image. At a different extreme, learning a prior over individual pixel colors would result in a 3-dimensional feature space where the distributions of different components are very likely to overlap, leaving our matting solution ambiguous. We avoid these two failure modes by having our discriminators operate on a multi-scale patch space using an architecture that resembles a multi-scale version of PatchGAN [Isola et al. 2017; Karnewar and Wang 2020].

In Stage 3, we train a set of discriminators adversarially along with the decomposition network \( N_D \). We freeze the environment layer, and set up one discriminator for the residual layer and one for the foreground layer. For discriminator \( D_i \), its positive examples \( X_i^+ \) are the augmented samples generated in Stage 2, and its negative examples \( X_i^- \) are color outputs from \( N_D \). The adversarial loss is:

\[
L_{\text{i,adv}} = \mathbb{E} \left[ \log \left( D_i(X_i^+) \right) \right] + \mathbb{E} \left[ \log \left( 1 - D_i(X_i^-) \right) \right]
\]

5.3 Likelihood
The adjusted framing of factor matting places less importance on the Bayesian likelihood than traditional matting. We reflect this in the weight of our reconstruction loss, which is given as:

\[
L_{\text{recon}} = \frac{1}{T} \sum_t ||V(t) - \text{Comp}(t)||_1
\]

where \( \text{Comp} \) is obtained by applying the “over” blending operator [Porter and Duff 1984] to the global ordering of layers taken from all components. Our default ordering places the background environment layer in the back, followed by the residual layer in the middle and the foreground layer in front.

5.4 Regularization
To further refine the solution, we impose regularization on the predicted alpha channels, and on an additional optical flow output produced by the decomposition network. Firstly, we use flow estimation as an auxiliary task for the decomposition network (with corresponding loss \( L_{\text{flow-est}} \), and use the learned flow to penalize temporal incoherence in color (\( L_{\text{RGB-warp}} \)) and alpha (\( L_{\text{warp}} \)). Secondly, we encourage alpha sparsity to prevent repeated content across multiple layers (\( L_{\text{alpha-reg}} \)). Thirdly, to speed up the initial convergence of foreground layer’s alpha channel, we encourage it to match the segmentation mask in the beginning of the training (\( L_{\text{mask-init}} \)). We turn off this loss once it is below a given threshold. The last two losses are very similar to those used in Omnimatte [2021], and we provide details in the supplemental materials.

5.4.1 Flow Estimation. For each layer \( \ell_i \), the network predicts a flow vector \( F_i(t, t + k) \) from time step \( t \) to \( t + k \). When \( k = 1 \), it is the most common case of consecutive frames. We use RAFT [Teed and Deng 2020] to estimate the “ground truth” flow vector \( F^\star(t, t + k) \). Current flow estimation algorithms assume that there is just a single mode of flow at each pixel. However, when complex interactions occur between layers, different flow fields may overlap; for example, a background pixel shadowed by a foreground object might exhibit some evidence of motion correlated with the foreground and some evidence correlated with the background. In such cases, the unimodal estimate is usually dominated by the layer with the most obvious motion. Therefore, rather than using the estimated ground truth to supervise all layers, we use it to supervise a single layer per pixel. At each pixel location, we find the layer for which the decomposition network assigns the flow vector closest to the ground truth, and penalize its deviation from the latter:

\[
L_{\text{flow-est}} = \frac{1}{T} \sum_t W(t, t + k) \cdot \min_{i} ||F_i(t, t + k) - F^\star(t, t + k)||_1
\]

where the pixel-wise \( \min \) pools the layer with the least error. \( W(t, t + k) \) is a spatial weighting map that reduces the contribution of pixels with inaccurate flow, measured based on standard left-right flow consistency and photometric errors [2021].

5.4.2 Temporal Coherence. An ideal factorization should produce layers that are temporally coherent. While the decomposition network already satisfies this aim to a degree, we found that regularizing by explicit loss terms improved the results:

\[
L_{\text{alpha-reg}} = \frac{1}{T N} \sum_i ||\text{Comp}(i) - \text{warp} (\text{Comp}(i(t + k)), F_i(t, t + k))||_1
\]

\[
L_{\text{RGB-warp}} = \frac{1}{T N} \sum_i ||\text{Comp}(i) - \text{warp} (\text{Comp}(i(t + k)), F_i(t, t + k))||_1
\]

The \( \text{warp} \) function warps layer \( \ell_i \) by predicted flow \( F_i(t, t + k) \). This loss is an extension of the one used in Omnimatte to pairs of frames that are not necessarily consecutive.
5.5 Optional Customization

For complex scenes, results can be further improved with very simple additional user inputs. Namely, many videos contain a range of cross-component interactions that take place over time, with some interactions creating more significant deformations than others. In such cases, applying our method first to a subset of frames with less complicated interactions can yield cleaner decompositions of those frames. These clean decompositions can then supply a subsequent optimization, performed over the entire video, with stronger marginal priors and high-confidence reference alpha mattes for regularization. To employ this strategy, the user only needs to select a smaller subset of video frames to optimize for first. Then, when training on the entire video in Stage 3, we add a $L_1$ loss penalizing differences from the initial solution on corresponding frames.

We can further increase the benefit of this strategy by manually cleaning a few frames of alpha produced in this first pass. These improvements can then be propagated to the rest of the video through our regularization terms. We only do the manual cleaning for the especially challenging simulated cushion videos, Puddle Splash in Fig. 1, and Trampoline (cropped) in Fig. 7.

6 AUGMENTATION STRATEGIES

Our augmentations are supposed to map the pixels of marginal priors to new patches that span the appearance of each layer under conditional interactions. These new patches are then used as the positive examples for training each discriminator. There are a few generic augmentations that we use by default for all layers, such as Gaussian blur, and addition of Gaussian noise. These transformations serve the dual purpose of approximating some possible conditional interactions and also providing general data augmentation to help our discriminator generalize to new inputs. In addition to these default transformations, we perform some more specific transformations designed to target conditional foreground and background interactions. Each of these transformations is applied at random with a pre-defined probability during training. This process is automatic, but could be further customized to interactions present in specific videos. We emphasize that our augmentations here are just examples of what one might use in this framework. Also note that because our conditional priors are patch-based, the images we generate during augmentation need only resemble our conditional interactions locally, and individual augmented frames need not preserve the global layout of our original scene (e.g., translations of the same image will produce the same patch samples). This makes it much easier to balance our need to produce examples that span the range of deformations caused by conditional interactions while also incorporating enough texture information to keep the foreground and background appearance distributions separate.

6.1 Foreground Transformations

Shadows. One common interaction is that of a foreground casting shadows on the background. We model this in our foreground augmentations by filling the non-foreground regions of our marginal prior images with dark gray scale colors. Similarly, in the case of a foreground that emits light, as in Fig. 11, we can fill these regions with slight variations on the color of illumination.

Reflections & Color Casts. These effects tend to impart colors from the foreground object onto the background, sometimes darkened by imperfect reflection. To create patches that span this sort of reflection, we can linearly interpolate between transformed copies of the object and patches of pure black.

Putting these together, in our default implementation the foreground positive examples are generated by:

1. Randomly choosing an input frame and extracting the object of interest using the provided mask,
2. (If the target video contains reflections) Randomly flipping, rotating, and scaling the color of the extracted region,
3. Filling empty regions uniformly with a random grey value, biased toward dark gray by default,
4. Randomly applying our generic augmentations.

At the patch level, this process yields a reasonable approximation of shadows, reflections, and color casts, which are the most common conditional effects of the foreground on other components.

6.2 Background Transformations

Our environment layer represents marginally likely parts of the background, forcing conditional background effects into our residual layer. We generate positive background samples by applying transformations to the background canvas generated in Stage 1.

Warping & Deformation. We use a grid-based warping function that applies a random translation to each vertex in a low-resolution regular grid, then upsample these translations to the resolution of our input video with a Gaussian kernel to generate a smooth displacement field and finally, apply this displacement field as a warp to a given image. We then randomly darken randomly selected Gaussian regions of the image to approximate bending and self-shading effects before finally apply our aforementioned generic augmentations. Fig. 4 shows some final outputs from the background augmentation pipeline. Please refer to the supplemental materials for examples on real videos.

7 RESULTS

We test our method on videos with and without complex interactions. Real-world videos featuring interacting elements lack ground truth counterfactual components, so we provide qualitative evaluations on these in Sec. 7.1, 7.3, and 7.4. In Sec. 7.2 and 7.5, we use datasets that do have ground truth decompositions to provide quantitative comparisons. We present results on videos from the DAVIS–2017 [Pont-Tuset et al. 2017] dataset, clips from in-the-wild videos on YouTube, and additional videos we recorded with standard consumer cellphones. In the supplemental materials, Table 1 describes the test videos we collected, Sec. 1 describes training details, and Sec. 2 shows more results. For the videos we collected, we use a rough binary mask obtained with a quick application of the Rotoscope tool in Adobe After Effects. Our training can be expedited by providing a clean frame featuring the static background without foreground objects or conditional effects. In such cases, Stage 1 is skipped. Acquiring this additional background image is often easy if the users records the input video themselves, so we adopt this practice for videos sourced from cellphone recordings.
7.1 Complex Scenes
Our test videos with complex interactions feature diverse foreground objects that cause a range of lighting effects including hard and soft shadows and reflections. The backgrounds feature varied opacity, rigidity, texture, shape, and motion.

7.1.1 Videos with Complex Interactions. By default, the compositing ordering of layers we use from back to front is: environment layer, residual layer, and finally foreground layer. However, as described in Sec. 5, the number and order of layers can vary for different scenes to accommodate the interleaving of content from different components. For instance, in Fig. 1 the transparent water splash appears both in front of and behind the child’s foot. To represent this, we use two alpha mattes for the residual layer that share the same color channels. Our composition order is then: environment layer, $A_{back}$ with residual layer color, foreground layer, $A_{front}$ with residual layer color. This lets our model represent parts of the splash that occur behind the feet as well as in front of the feet. This task is particularly challenging due to the transparency of the water; the discriminator must learn the flow, color, and water texture from the surroundings and apply this knowledge to the region where the water and foot overlap. To our knowledge, no other methods are designed to handle such complex interactions.

The interaction in Ski (second row in Fig. 5) includes the skier’s shadow and flying snow caused by the skier landing on the mountainside. Since the scene is primarily composed of shades of white or gray, the foreground (shadow) and background (flung-up snow) have significant color overlap. Nonetheless, FactorMatte still produces a clean separation due to the use of motion cues.

Conditional interactions are relatively rare among examples used in previous matting works, presumably because most methods were not designed to address such interactions. One partial exception is the work of Omnimate [2021] and its predecessor [Lu et al. 2020], which include examples with shadows, reflections, and some limited deformations. Fig. 7 shows a crop from the trampoline video, one of the most difficult cases for factor matting that appeared in their work. If we consider the quality of the re-compositing results, both Omnimate and FactorMatte produce errors around the trampoline, yet our method produces an overall cleaner matte that can be used to create a counterfactual video with the jumper removed.

The Kite-Surf video (third row in Fig. 5) depicts a foreground surfer and features significant foreground-background interaction due to the action of surfing. Waves appear in all frames as conditional effects, and the regions of them that can be described by a homography are captured in the environment layer during Stage 1. Then the residual layer positive examples generated from those marginal priors also contain some wave colors and textures, which obviates the need to manually simulating waves from scratch.

7.1.2 Background Parallax. Conditional effects stemming from interactions with the foreground are just one way that even advanced variations of background subtraction can fail. Other causes include spatially-varying parallax caused by camera translation or changes in scene lighting, which are common in real videos. One benefit of our conditional priors is that they help to factor these changes into foreground and background effects. For example, the Hike video in Fig. 5 was included as a failure case in the Omnimate paper [2021]. This video shows a person walking in front of a mountain, where camera translation and a significant range of scene depths lead to spatially-varying parallax that cannot be approximated with a simple homography. This causes Omnimate to place much of the mountain’s visual changes in the foreground. In contrast, our method correctly assigns the mountain’s visual changes to the residual layer and gives a clean foreground containing only the hiker and his shadow.

In Fig. 6, we show another important difference in the color layers. The goal of factor matting is to separate the appearance of a scene into components that each represent likely contributions of different content to new scenes. While Omnimate also factors a scene into layers, over-optimizing for the reconstruction loss of such a factorization often results in foreground color channels that duplicate content from the background. This duplication ensures that the reconstruction loss can be satisfied equally well by multiple layers, which lets the network reduce overall reconstruction loss by introducing errors into the foreground alpha channel. FactorMatte prevents this by using explicit priors on the local appearance of each layer to disentangle foreground and background content.

7.2 Classical Video Matting
While FactorMatte is designed to address videos featuring complex cross-component interactions, we find it also does well on scenes without such interactions. We evaluate our method on VideoMatting108 [Zhang et al. 2021], a very challenging dataset that contains
moving backgrounds and various foreground objects such as humans, animals, fluffy toys, cloth, plants, and smoke. We compare with RVM [Lin et al. 2022], BGMv2 [Lin et al. 2021], DIM [2017], IndexNet [2019], GCA [2020], and two variations of TCVOM [2021]. In Table 1, we compute SSDA (average sum of squared difference), MSE (mean squared error), MAD (mean absolute difference), dtSSD (mean squared difference of direct temporal gradients), and MESS-Ddt (mean squared difference between the warped temporal gradient) [Erofeev et al. 2015] on the validation subset provided by the dataset.

We note that the scores of RVM [2022] and BGMv2 [2021] may not reflect the true value of these methods, as we are evaluating them here in scenarios that differ from their intended use. BGMv2 is designed assuming a fixed background is given as input; here we average its performance when giving the first, 100-th, and 200-th background images as input. RVM is trained to work on videos with humans as foreground subjects, which is not true for all of the test data. Both of these methods perform much better when their corresponding assumptions hold.

FactorMatte and Omnimatte are both optimized per-video without training on any external matting datasets. In some sense, we can think of these approaches as exploring how well we can do on matting when limited primarily to evidence contained in each input video. Therefore, one would expect that some methods trained on outside data to perform better on classical matting, such as TCVOM in Table 1. Additionally, such classical matting methods always contain objectives explicitly encouraging accurate foreground alpha mattes, e.g. the alpha prediction training loss. However, our method does not assume such ground truth alpha values are provided. Notably, there are currently no labeled datasets for the type of complex interactions that motivate our work, and even collecting such data for real scenes remains an open problem.
Trampoline (cropped). In this video, the person jumps onto the trampoline and bounces back into the air (the full version can be found in [2020]). The first row shows a frame where the person is in the air, and the second row shows a frame with the trampoline deformed due to physical interactions. Our foreground is less contaminated by background elements, and our background features more of the trampoline’s deformation.

We also evaluate on real videos captured by consumer phones fixed on a tripod. Fig. 8 compares our results to those of Omnimate and Background Matting (BGM) [Sengupta et al. 2020] on the test clips from BGMv2 [Lin et al. 2021]. Due to memory constraints, BGM fixes the input resolution to 512×512. We stick to the same resolution, and also due to memory limits, we are unable to compare on HD inputs. We follow BGM’s original practice to use DeepLabv3+ [Chen et al. 2018] for mask extraction, and the static background image that they require is leveraged to generate higher-quality positive examples for the residual layer in our pipeline. When generating foreground layer positive examples, we erode the segmentation masks to avoid background colors. As in Fig. 8, FactorMatte recovers the most hair details among all three.

7.3 After Effects Plugin & Re-Compositing

FactorMatte was explicitly designed with re-compositing applications in mind. To facilitate these, we developed a plug-in for Adobe After Effects that loads the output of FactorMatte into a hierarchy of compositions for editing. These compositions include layers of each scene, as well as the result of binary operators between layers and input masks. The plug-in offers several controls and visualization tools, including template compositions with empty layers that users can drag replacement foreground or background objects onto for fast re-composition. Fig. 9 shows some of the previous examples re-composited with virtual contents using our plug-in. A version of our plugin is linked from our project website.

Table 1. Evaluation on Classical Matting. Following the practice of TCVOM [Zhang et al. 2021], we use the middle width when dilating the ground truth alpha images to generate tri-maps. We use the models released from each method’s official repository, and evaluate on the first 250 frames of each video, down-sampled to 352×512. We mark the ranking of our method and the most comparable one, Omnimate, on the top right of each value (ranking index starts from 1).

| Method          | SSDA ↓ | MSE ↑ | MAD ↓ | dtSSD ↓ | MESSDdt ↓ |
|-----------------|--------|-------|-------|---------|-----------|
| RVM             | 68.02  | 207.41| 272.68| 14.31   | 1.48      |
| BGMv2           | 50.23  | 122.92| 184.46| 17.16   | 1.26      |
| IndexNet        | 34.25  | 31.35 | 90.08 | 14.27   | 1.18      |
| Omnimate        | 33.82  | 65.80 | 132.49| 10.59   | 0.62      |
| DIM             | 31.07  | 26.04 | 82.33 | 12.43   | 0.77      |
| GCA             | 26.26  | 20.35 | 66.77 | 11.36   | 0.56      |
| FactorMatte     | 25.07  | 18.99 | 67.93 | 8.22    | 0.39      |
| TCVOM (GCA)     | 18.05  | 12.29 | 48.83 | 9.02    | 0.23      |
| TCVOM (FBA)     | 16.15  | 9.92  | 41.64 | 7.91    | 0.17      |

7.4 Additional Video Editing Effects

The output of FactorMatte can also be combined with other methods for downstream applications. One good example is object removal. While the counterfactual background video produced by FactorMatte already is somewhat akin to the result of object removal, inpainting is a nice side effect, rather than a major goal, of our method, and it is done based only on information in the video itself.

On the other hand, state-of-the-art texture inpainting methods are
Fig. 9. **Re-Compositions.** In (a) and (d) we replace the foreground while preserving the background conditional effects. In (b) we add additional hand-drawn layers between foreground and background layers. In (c) we replace the background. Please see our website for full results. The added materials in (a), (b) and (d) are from the Internet; the background of (c) uses the cover of album *Abbey Road* of "The Beatles".

Fig. 10. **Inpainting.** We evaluate different methods on video inpainting, using the masks shown in blue. For Omnimatte (b) and FactorMatte (d), the masks are acquired by binarizing the predicted alpha channel with a fixed threshold. (c) uses the original segmentation mask. For easier comparison, we show an input frame of the static cushion without the stick in (e). The result (h) using FactorMatte alpha successfully removes the shadow while preserving the deformation near the bottom of the cushion.

specialized for this task and can draw from large datasets for training. Such methods require an input mask to indicate the region to be inpainted, and therefore the quality of this mask can greatly influence the results. In Fig. 10, we feed masks generated from different matting methods as input to a recent state-of-the-art video inpainting pipeline, E2FGVI [Li et al. 2022]. Simple segmentation masks tend to leave correlated effects like shadows in the scene (Fig. 10g), while masks from background subtraction–based methods lead to removal of most interaction effects, including deformations of the background (Fig. 10f). The mask from FactorMatte contains the object and its shadow but not the cushion, thus leading to the most plausible invisible result as shown in Fig. 10h.

Fig. 11. **Color Pop.** We can apply color editing to the foreground layer. While FactorMatte successfully changes the color of the light cast by the flashlight to red, Omnimatte gives many artifacts.

Fig. 12. **Freezing Time.** By stacking the tennis player with their shadow at different time steps, we can make the moving camera as if fixed and visualize the motion trajectory. This video is from DAVIS–2017 [2017].

Improving the quality of alpha mattes also enables us to shift the color or timing of components within a video more aggressively than previous methods. The video in Fig. 11 shows a person waving a flashlight over a textured surface. Here, we change the color of the flashlight’s beam by adjusting the foreground color layer to be more red outside of the input foreground mask.

Re-timing tasks are more forgiving of matting errors because mistakes in the alpha channel of static scene content will not typically lead to major artifacts. Therefore, prior methods can already produce high-quality results for re-timing applications, and FactorMatte maintains this ability as shown in Fig. 12.

7.5 Ablations
Quantitative ablations are difficult for real-world results due to the lack of available ground truth, but we can perform qualitative evaluations on real videos (e.g., Fig. 15) and both quantitative and qualitative evaluations on synthetic videos (e.g., Fig. 13 and 14). For synthetic data we rendered a dataset of simulated videos in Blender, each featuring interactions between a solid object and a deformable one. The scenes vary in object shape, texture, color, and interaction movements. Table 2 reports the numerical evaluation on 21 video results each containing 160-170 frames. Qualitatively, when \( L_{flow-est} \) is removed (Fig. 13d), the decomposition may fail to group some pixels that are primarily related through their correlated flow. With only color information, the network is more prone to mistakes that are characteristic of image-only matting. This is particularly
Table 2. Quantitative Ablations Results. We show quantitative results with different losses removed. For each result, we average per-pixel MAD on the foreground alpha, and MSE on the counterfactual background composited by the residual and the environment layer. All results are acquired without residual augmentations as they do not improve the model performance. The leftmost column ("FactorMatte") shows the result of our method, and the last column shows the result of using Omnimatte with an additional "residual layer" included to help capture deformations in the background. Each of the other columns shows the result of removing one element (listed in the header) from FactorMatte.

|                  | FactorMatte | Residual D | Foreground D | Patch Arch. | Augmentations | Residual L | Omnimatte w/ Residual L |
|------------------|-------------|------------|--------------|-------------|---------------|------------|------------------------|
| MSE              | 0.005       | 0.006      | 0.008        | 0.007       | 0.006         | 0.007      | 0.018                  |
| MAD              | 0.004       | 0.003      | 0.084        | 0.004       | 0.010         | 0.261      | 0.010                  |

Fig. 13. Ablations. (a) shows the input mask (top) and frame (bottom) for a frame from the Purple Monkey video. For the rest of this figure, the first row shows the foreground layer’s alpha channel for various ablation experiments, and the second row shows the counterfactual background component generated by compositing the residual layer and the environment layer. (b) Simulated ground truth; (c) our full method; (d) results when turning off the flow estimate loss; (e) results when turning off the consistency loss; (f) results when turning off the residual layer discriminator.

Flow information alone is insufficient for videos with complex interactions. In Fig. 13e the alpha consistency regularization is turned off, and we find that the foreground layer alpha flickers across frames. Without the coherence constraint, correct predictions before and after interactions no longer help as much with the more difficult factorization of interaction frames.

If there is a complex layer that lacks a discriminator, the model can assign spurious contents to that layer rejected by other discriminators. In Fig. 13f, we turn off the residual discriminator. Since a part of the monkey is refused by the foreground discriminator, the background counterfactual component contains it as an artifact.

Fig. 14 shows background components extracted from synthetic video of a green ball bouncing on and off a densely-textured cushion. We compare our augmented multi-scale patch-based architecture with two non-patch-based alternatives: one that follows our architecture with fully connected layers (row 2), and another that uses ResNet18 [He et al. 2016] (row 3). In both cases we see artifacts and the inclusion of the ball’s shadow in the background component. We also see more artifacts when we turn off foreground augmentation (row 4), training the foreground discriminator only on the marginal prior for that component. Without foreground augmentation, the balls shadow does not match the appearance model of our foreground layer and is pushed to the residual component. Note that our full result here, while preferable to the others, is also not perfect, as we do not have any observations of the true deformed state of the cushion under the ball.

Fig. 15 shows how augmentation on the residual layer improves results. The augmentations used in this example included grid-based warping, Gaussian blur, and Gaussian random noise. We noticed
that residual augmentation in this result is more important than the one shown in Fig. 14. We believe this is because the cushion in Fig. 14 has a repeating pattern, so different parts of the cushion serve as augmented versions of that pattern even within a single frame. This agrees with our observation that the patch-based discriminator is important.

8 LIMITATIONS AND FUTURE WORK
At an architectural level, the main addition of FactorMatte relative to previous work is an augmented patch-based appearance prior. The use of a patch-based prior makes spanning conditional interactions tractable, but the best results still require some amount of hand-tuning to achieve. In particular, for the more challenging real-world examples, we found that parameter tuning (e.g., the choice of augmentations, or taking the steps described in Sec 5.5) can have a notable impact on the quality of results. Also, while we perform well on classical matting tasks, our results on such tasks fall behind some methods that train on large labeled datasets. These limitations may both be difficult to overcome without incorporating priors learned from external data, as the problem we address is fundamentally very under-constrained. One way to understand this is through a comparison with classical Bayesian matting. If we consider a scene with similar foreground and background colors, classical Bayesian matting will often fail because the distributions of these colors overlap. Our patch-based prior addresses this by increasing the dimensionality of each distribution, making them easier to separate. Our augmentations then provide a way to balance this higher dimensionality against a need to generalize each distribution to conditional interactions. Overall, this is a powerful strategy, but it still relies on observable differences between each component. We have no way of inferring completely unseen content, or factoring conditional interactions that result in unexpected changes of appearance. For these scenarios, further training on external data or incorporating more user input are promising directions.

The most notable practical limitation of our current implementation is runtime. As our method contains multiple training stages and additional networks, our runtime approximately ranges from 2–5 times that of Omnimatte, which is itself much slower than most state of the art video matting techniques. Both Omnimatte and our work focus on optimizing a simple set of well-chosen priors without training on external data. This is a good way to explore the merits of those priors, and a useful strategy for exploring scenarios where real-world labeled training data is particularly difficult to obtain.

9 CONCLUSION
There are two major takeaways from our work. The first is a reframing of the video matting problem in terms of counterfactual video synthesis, which is a particularly useful framing for downstream re-compositing tasks. By solving for counterfactual components that answer questions of the form “what would this content look like if we froze time and separated it from the rest of the scene?” we end up with a flexible and reusable factorization of video. Our Bayesian formulation of the problem provides a nice explanation for how this formulation is more general than the typical assumption of independent layers made in classical matting approaches. The second significant takeaway is the use of an augmented patch-based prior on appearance, which offers a flexible way to combine representative sample pixels with simple transformations to span the appearance of conditional interactions. Finally, we showed how these observations can help generalize video matting to scenes with complex interactions and enable new re-compositing applications. We believe this work points to a new direction for video matting with exciting applications in video editing and analysis.

ACKNOWLEDGEMENTS
Wenqi Xian was supported on a fellowship from Microsoft.

REFERENCES
Xue Bai, Jue Wang, and David Simons. 2011. Towards temporally-coherent video matting. In International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications.
Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. 2009. Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. 28, 3 (2009).
Olivier Barnich and Marc Van Droogenbroeck. 2010. ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. 20, 6 (2010).
Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proc. European Conference on Computer Vision (ECCV).
QiFeng Chen, Dingzeyu Li, and Chi-Keung Tang. 2013. KNN matting. Trans. Pattern Anal. Mach. Intell. 35, 9 (2013).
Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. 2016. Natural image matting using deep convolutional neural networks. In Proc. European Conference on Computer Vision (ECCV).
Inchang Choi, Minhaeng Lee, and Yu-Wing Tai. 2012. Video matting using multi-frame nonlocal matting laplacian. In Proc. European Conference on Computer Vision (ECCV). 540–553.
Yung-Yu Chuang, Aseem Agarwal, Brian Curless, David A Salesin, and Richard Szeliski. 2002. Video matting of complex scenes. In SIGGRAPH.

Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. 2001. A bayesian approach to digital matting. In Proc. Computer Vision and Pattern Recognition (CVPR), Vol. 2.

Mikhail Efroymson, Yuriy Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Yue Wang. 2015. Perceptually Motivated Benchmark for Video Matting. In BMVC. 99–1.

Marco Forte and François Pitié. 2020. F. B. Alpha Matting. In Proc. Computer Vision and Pattern Recognition (CVPR).

Qiqi Hou and Feng Liu. 2019. Context-aware image matting for simultaneous foreground and alpha estimation. In Proc. Int. Conf. on Computer Vision (ICCV).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR.

Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. 2004. Poisson Matting. In Proc. Computer Vision and Pattern Recognition (CVPR).

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In Proc. European Conference on Computer Vision (ECCV).

Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. 2010. Temporally coherent video matting. In Proc. Int. Conf. on Computer Vision (ICCV), Vol. 2.

Richard J Qian and M Ibrahim Sezan. 1999. Video background replacement without a trimap. In Proc. Int. Conf. on Computer Vision (ICCV), Vol. 2.

Yaoyi Li and Hongtao Lu. 2020. Natural image matting via guided contextual attention. In ACM Trans. Graph., Vol. 39, No. 6. 17562–17571.

Long Ang Lim and Hacer Yalim Keles. 2010. Foreground segmentation using convolutional neural networks for multiscene feature encoding. Pattern Recognition Letters 112 (2018). 17562–17571.

Hongjie Tang, Yugao Aksos, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. 2019. Learning-based sampling for natural image matting. In Proc. Computer Vision and Pattern Recognition (CVPR).

Omnimatte: Associating objects and their effects in video. In Proc. Int. Conf. on Computer Vision (ICCV), Vol. 2. 1015721.

Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. 2021. Deep video matting via spatio-temporal alignment and aggregation. In Proc. Computer Vision and Pattern Recognition (CVPR).

Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. 2022. Modnet: Perceptually Motivated Benchmark for Video Matting. In Proc. Computer Vision and Pattern Recognition (CVPR).

William T Freeman, and Michael Rubinstein. 2021. Omnimatte: Associating objects and their effects in video. In Proc. Int. Conf. on Computer Vision (ICCV). 17571–17580.

Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. 2022. Towards an end-to-end framework for flow-guided video inpainting. In CVPR. 17562–17571.

J. Wang and M.F. Cohen. 2005. An iterative optimization approach for unified image segmentation and matting. In Proc. Int. Conf. on Computer Vision (ICCV), Vol. 2. https://doi.org/10.1109/ICCV.2005.37

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. 2017. Deep image matting. In CVPR. 2970–2979.

Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. 2021. Deformable Sprites for Unsupervised Video Decomposition. In Proc. Computer Vision and Pattern Recognition (CVPR).

Yunke Zhang, Chi Wang, Xiaomiao Cui, Peiran Ren, Xuanxong Xie, Xian-Sheng Hua, Hujun Bao, Qixing Huang, and Wener Xu. 2021. Attention-guided Temporally Coherent Video Object Matting. In ACMMM. 5128–5137.