Background Matting: The World is Your Green Screen

Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman

University of Washington

Figure 1: Using a handheld smartphone camera, we capture two images of a scene, one with the subject and one without. We employ a deep network with an adversarial loss to recover alpha matte and foreground color. We composite the result onto a novel background.

Abstract

We propose a method for creating a matte – the per-pixel foreground color and alpha – of a person by taking photos or videos in an everyday setting with a handheld camera. Most existing matting methods require a green screen background or a manually created trimap to produce a good matte. Automatic, trimap-free methods are appearing, but are not of comparable quality. In our trimap free approach, we ask the user to take an additional photo of the background without the subject at the time of capture. This step requires a small amount of foresight but is far less time-consuming than creating a trimap. We train a deep network with an adversarial loss to predict the matte. We first train a matting network with supervised loss on ground truth data with synthetic composites. To bridge the domain gap to real imagery with no labeling, we train another matting network guided by the first network and by a discriminator that judges the quality of composites. We demonstrate results on a wide variety of photos and videos and show significant improvement over the state of the art.

1. Introduction

Imagine being able to easily create a matte — the per-pixel color and alpha — of a person by taking photos or videos in an everyday setting with just a handheld smartphone. Today, the best methods for extracting ("pulling") a good quality matte require either a green screen studio, or the manual creation of a *trimap* (foreground/background/unknown segmentation), a painstaking process that often requires careful painting around strands of hair. Methods that require neither of these are beginning to appear, but they are not of comparable quality. Instead, we propose taking an additional photo of the (static) background just before or after the subject is in frame, and using this photo to perform *background matting*. Taking one extra photo in the moment requires a small amount of foresight, but the effort is tiny compared to creating a trimap after the fact. This advantage is even greater for video input. Now, the world is your green screen.

We focus on a method that is tuned to human subjects. Still, even in this setting — pulling the matte of a person given a photo of the background — the problem is ill-posed and requires novel solutions.

Consider the compositing equation for image I given foreground F, background B, and mixing coefficient α: $I = \alpha F + (1-\alpha)B$. For color images and scalar α, and given B, we have four unknowns (F and α), but only three observations per pixel (I). Thus, the background matting problem is underconstrained. Background/foreground differences provide a signal, but the signal is poor when parts of the person are similar in color to the background. Furthermore, we do not generally have an image of the ideal background: the subject can cast shadows and cause reflections not seen in the photo taken without the subject, and exact, pixel-level alignment with no resampling artifacts between handheld capture of two photos is generally not attainable. In effect, rather than the true B that produced

I, we have some perturbed version of it, B'. Finally, we can build on person segmentation algorithms to make the problem more tractable to identify what is semantically the foreground. However current methods, exhibit failures for complex body poses and fine features like hair and fingers.

Given these challenges and recently published successes in solving matting problems, a deep learning approach is a natural solution. We propose a deep network that estimates the foreground and alpha from input comprised of the original image, the background photo, and an automatically computed soft segmentation of the person in frame. The network can also utilize several frames of video, useful for bursts or performance capture, when available. However, the majority of our results, including all comparisons to single-image methods, do not use any temporal cues.

We initially train our network on the Adobe Matting dataset [35], comprised of ground truth mattes that can be synthetically composited over a variety of backgrounds. In practice, we found the domain gap between these synthetic composites and real-world images did not lead to good results using standard networks. We partially close this gap in two ways: by augmenting the dataset and by devising a new network — a "Context Switching Block" — that more effectively selects among the input cues. The resulting mattes for real images can still have significant artifacts, particularly evident when compositing onto a new background. We thus additionally train the network in a self-supervised manner on real unlabelled input images using an adversarial loss to judge newly created composites and ultimately improve the matting process.

Our method has some limitations. First, we do require two images. Trimap-based methods arguably require two images as well for best results – the trimap itself is a hand-made second image – though they can be applied to any input photo. Second, we require a static background and small camera motion; our method would not perform well on backgrounds with people walking through or with a camera that moves far from the background capture position. Finally, our approach is specialized to foregrounds of (one or more) people. That said, person matting without big camera movement in front of a static background is, we argue, a very useful and not uncommon scenario, and we deliver state-of-the-art results under these circumstances.

Our contributions include: • The first trimap-free automatic matting algorithm that utilizes a casually captured background. • A novel matting architecture (Context Switching Block) to select among input cues. • A self-supervised adversarial training to improve mattes on real images. • Experimental comparisons to a variety of competing methods on wide range of inputs (handheld, fixed-camera, indoor, outdoor), demonstrating the relative success of our approach. Our code and data is available at http://github.com/senguptaumd/Background-Matting.

2. Related Work

Matting is a standard technique used in photo editing and visual effects. In an uncontrolled setting, this is known as the "natural image matting" problem; pulling the matte requires solving for *seven* unknowns per pixel (F, B, α) and is typically solved with the aid of a trimap. In a studio, the subject is photographed in front of a uniformly lit, constant-colored background (e.g., a green screen); reasonable results are attainable if the subject avoids wearing colors that are similar to the background. We take a middle ground in our work: we casually shoot the subject in a natural (non-studio) setting, but include an image of the background without the subject to make the matting problem more tractable. In this section, we discuss related work on natural image matting, captured without unusual hardware.

Traditional approaches. Traditional (non-learning based) matting approaches generally require a trimap as input. They can be roughly categorized into sampling-based techniques and propagation-based techniques. Sampling-based methods [11, 9, 14, 28, 32, 33, 2] use sampling to build the color statistics of the known foreground and background, and then solve for the matte in the 'unknown' region. Propagation-based approaches [6, 17, 19, 20, 30, 13, 15] aim to propagate the alpha matte from the foreground and the background region into the 'unknown' region to solve the matting equation. Wang and Cohen [34] presents a nice survey of many different matting techniques.

Learning-based approaches. Deep learning approaches showed renewed success in natural image matting, especially in presence of user-generated trimaps. Some methods combine learning-based approaches with traditional techniques, e.g., KNN-matting [29, 7]. Xu *et al.* [35] created a matting dataset with real mattes and composited over a variety of backgrounds and trained a deep network to predict the alpha matte; these results were further improved by Lutz *et al.* [22] using an adversarial loss. Recently Tang *et al.* [31] proposed a hybrid of a sampling-based approach and learning to predict the alpha matte. Lu *et.al* [21] proposed a new index-guided upsampling and unpooling operation that helps the network predict better alpha mattes. Cai *et al.* [3] showed robustness to faulty user-defined trimaps. All of these methods only predict the alpha matte and not the foreground, leaving open the (non-trivial) problem of recovering foreground color needed for composites. Recently Hou *et al.* [16] introduced Context-Aware Matting (CAM) which simultaneously predicts the alpha and the foreground, thus solving the complete matting problem, but is not robust to faulty trimaps. In contrast to these methods (and the traditional approaches), our work jointly predicts alpha and foreground using an image of the background instead of a trimap.

Recently, researchers have developed algorithms that perform matting without a trimap, focusing mostly on hu-

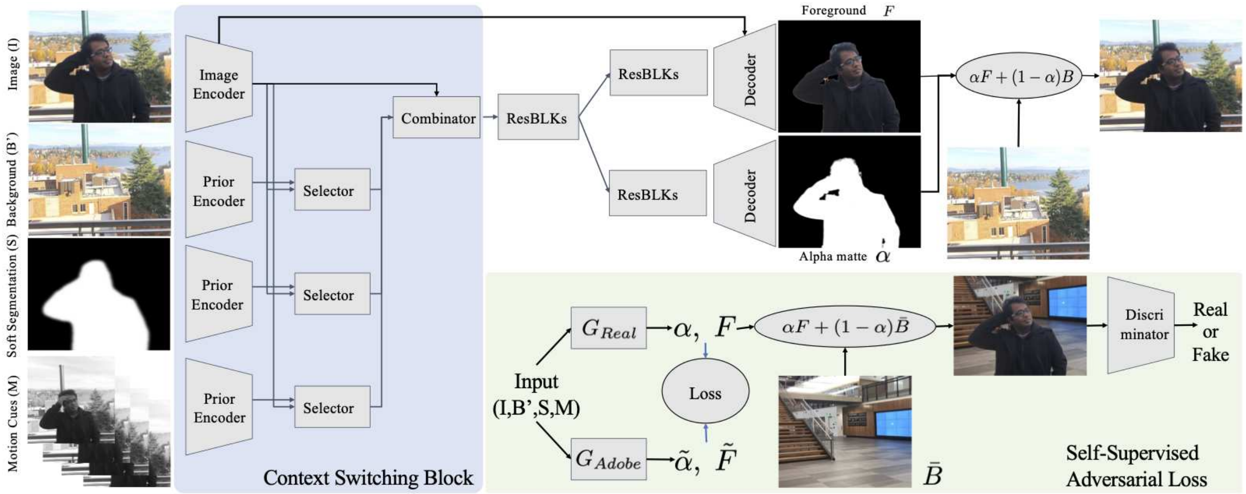

Figure 2: **Overview of our approach.** Given an input image I and background image B', we jointly estimate the alpha matte α and the foreground F using soft segmentation S and motion prior M (for video only). We propose a Context Switching Block that efficiently combines all different cues. We also introduce self-supervised training on unlabelled real data by compositing into novel backgrounds.

mans (as we do). Aksoy *et.al.* [1] introduced fully automatic semantic soft segmentation for natural images. In [37, 29] the authors perform portrait matting without trimap, utilizing segmentation cues. Trimap-free matting has also been extended to handle whole bodies in [36, 5]. These methods aim to perform trimap prediction, followed by alpha prediction. Our work is also human-focused; we compare our approach with the recent state-of-the-art automatic human matting algorithm [36] and obtain significantly better performance with the aid of the background image.

Matting with known natural background. Difference matting proposed by Qian and Sezan [25] attempts to solve matting with a natural background by simple background subtraction and thresholding but is very sensitive to the threshold and produces binary mattes. Similarly, change detection via background subtraction [24, 10] generally does not produce alpha mattes with foreground and considers shadows to be part of the foreground. Some traditional approaches like Bayesian matting [9] and Poisson matting [30, 12] can handle known background in their framework, but additionally require trimaps.

Video Matting. Researchers have also focused on video-specific methods. Chuang *et.al.* [8] extended Bayesian Matting to videos by utilizing the known background and optical flow, requiring trimaps for keyframes. Flow-based temporal smoothing can be used [18, 27] (again with trimaps) to encourage temporal coherence.

3. Our Approach

The input to our system is an image or video of a person in front of a static, natural background, plus an image of just the background. The imaging process is easy, just requiring the user to step out of the frame after the shot to capture the background, and works with any camera with a setting to lock the exposure and focus (e.g., a smartphone camera). For handheld capture, we assume camera motion is small and align the background to a given input image with a homography. From the input, we also extract a soft segmentation of the subject. For video input, we can additionally utilize nearby frames to aid in matting.

At the core of our approach is a deep matting network G that extracts foreground color and alpha for a given input frame, augmented with background, soft segmentation, and (optionally nearby video frames), and a discriminator network D that guides the training to generate realistic results. In Section 3.1, we describe the matting network, which contains a novel architecture – a "Context-switching block" – that can combine different input cues selectively. We first train a copy of this network G_{Adobe} with supervision using the Adobe Matting Dataset [35]. We use known foreground and alpha mattes of non-transparent objects, which are then composited over a variety of backgrounds (i.e., real source images, but synthetic composites). Our matting network, along with some data augmentation, help overcome some of the domain gap between the synthetically composited imagery and real data that we later capture with a consumer camera (e.g., a smartphone).

In Section 3.2, we describe a self-supervised scheme to bridge the domain gap further and to generally improve the matting quality. The method employs an adversarial network comprised of a separate copy of the deep matting network, G_{Real}, that tries to produce a matte similar to the output of G_{Adobe} and a discriminator network D that scores the result of compositing onto a novel background as real or fake. We train G_{Real} and D jointly on real inputs, with supervision provided by (the now fixed) G_{Adobe} network applied to the same data.

3.1. Supervised Training on the Adobe Dataset

Here we describe our deep matting network, which we first train on the Adobe Matting Dataset, restricted to the subset of non-transparent objects. The network takes as input an image I with a person in the foreground, an image of the background B' registered to I (as noted earlier, B' is not the same as the true B with subject present), a soft segmentation of the person S, and (optionally for video) a stack of temporally nearby frames M, and produces as output a foreground image F and alpha matte α. To generate S, we apply person segmentation [4] and then erode (5 steps), dilate (10 steps), and apply a Gaussian blur ($\sigma = 5$). When video is available, we set M to be the concatenation of the two frames before and after I, i.e., $\{I_{-2T}, I_{-T}, I_{+T}, I_{+2T}\}$ for frame interval T; these images are converted to grayscale to ignore color cues and focus more on motion cues. In the absence of video, we simply set M to $\{I, I, I, I\}$, also converted to grayscale. We denote the input set as $X \equiv \{I, B', S, M\}$. The network with weight parameters θ thus computes:

$$(F, \alpha) = G(X; \theta). \quad (1)$$

In designing and training the network, the domain gap between the Adobe dataset and our real data has proven to be a significant driver in our choices as we describe below.

A natural choice for G would be a residual-block-based encoder-decoder [38] operating on a concatenation of the inputs $\{I, B', S, M\}$. Though we would expect such a network to learn which cues to trust at each pixel when recovering the matte, we found that such a network did not perform well. When training on the Adobe synthetic-composite data and then testing on real data, the resulting network tended to make errors like trusting the background B' too much and generating holes whenever F was too close in color; the network was not able to bridge the domain gap.

Instead, we propose a new Context Switching block (CS block) network (Figure 2) to combine features more effectively from all cues, conditioned on the input image. When, e.g., a portion of the person matches the background, the network should focus more on segmentation cue in that region. The network has four different encoders for I, B', S, and M that separately produce 256 channels of feature maps for each. It then combines the image features from I with each of B', S and M separately by applying 1x1 convolution, BatchNorm, and ReLU ('Selector' block in Fig. 2), producing 64-channel features for each of the three pairs. Finally, these three 64-channel features are combined with the original 256-channel image features with 1x1 convolution, BatchNorm, and ReLU (the 'Combinator' block in Fig. 2) to produce encoded features which are passed on to the rest of the network, consisting of residual blocks and decoders. We observe that the CS Block architecture helps to generalize from the synthetic-composite Adobe dataset to real data (Figure 4). More network architecture details are provided in the supplementary material.

We train the network with the Adobe Matting dataset [35] which provides 450 ground truth foreground image F^* and alpha matte α^* (manually extracted from natural images). We select the subset of 280 images corresponding to non-transparent objects (omitting, e.g., objects made of glass). As in [35], we can compose these foregrounds over known backgrounds drawn from the MS-COCO dataset, augmented with random crops of varying resolutions, re-scalings, and horizontal flips. These known backgrounds B would not be the same as captured backgrounds B' in a real setting. Rather than carefully simulate how B and B' might differ, we simply perturbed B to avoid training the network to rely too much on its exact values. In particular, we generated each B' by randomly applying either a small gamma correction $\gamma \sim \mathcal{N}(1, 0.12)$ to B or adding gaussian noise $\eta \sim \mathcal{N}(\mu \in [-7, 7], \sigma \in [2, 6])$ around the foreground region. Further, to simulate imperfect segmentation guidance S we threshold the alpha matte and then erode (10-20 steps), dilate (15-30 steps) and blur ($\sigma \in [3, 5, 7]$) the result. For the motion cue M, we applied random affine transformations to foreground+alpha before compositing onto the background, followed by conversion to grayscale. To compute I and M we used the compositing equation with B as the background, but we provided B' as the input background to the network.

Finally, we train our network $G_{\text{Adobe}} \equiv G(\cdot; \theta_{\text{Adobe}})$ on the Adobe dataset with supervised loss:

$$\begin{aligned}\min_{\theta_{\text{Adobe}}} & E_{X \sim p_X}[\|\alpha - \alpha^*\|_1 + \|\nabla(\alpha) - \nabla(\alpha^*)\|_1 \\ & + 2\|F - F^*\|_1 + \|I - \alpha F - (1-\alpha)B\|_1],\end{aligned} \quad (2)$$

where $(F, \alpha) = G(X; \theta_{\text{Adobe}})$, and the gradient term on α encourages sharper alpha mattes [36].

3.2. Adversarial Training on Unlabelled Real data

Although our proposed Context Switch block (CS block) combined with data augmentation significantly helps in bridging the gap between real images and synthetic composites created with the Adobe dataset, it still fails to handle all difficulties present in real data. Theses difficulties include (1) traces of background around fingers, arms, and hairs being copied into the matte; (2) segmentation failing; (3) significant parts of the foreground color matching the background color; (4) misalignment between the image and the background (we assume only small misalignment). To handle these cases, we aim to learn from unlabelled, real data (real images + backgrounds) with self-supervision.

The key insight is that significant errors in the estimated matte typically result in unrealistic composites over novel backgrounds. For example, a bad matte might contain a chunk of the source background, which, when composited over a new background, will have a piece of the original background copied over the new background, a major visual artifact. Thus, we can train an adversarial discriminator to distinguish between fake composites and (already captured) real images to improve the matting network.

The matting network ($G_{\text{Real}} \equiv G(\cdot; \theta_{\text{Real}})$) and discriminator network D can be trained end-to-end based on just a standard discriminator loss. However, G_{Real} could settle on setting $\alpha = 1$ everywhere, which would result in simply copying the entire input image into the composite passed to D. This solution is "optimal" for G_{Real}, since the input image is indeed real and should fool D. Initializing with G_{Adobe} and fine-tuning with a low learning rate (was necessary for stable training with a discriminator) is not very effective. It does not allow significant changes to network weights needed to generate good mattes on real data.

Instead, we use G_{Adobe} for teacher-student learning. In particular, for a real training image I and associated inputs comprising X, we obtain $(\tilde{F}, \tilde{\alpha}) = G(X; \theta_{\text{Adobe}})$ to serve as "pseudo ground-truth". We can now train with an adversarial loss *and* a loss on the output of the matting network $G(X; \theta_{\text{Real}})$ when compared to "pseudo ground-truth", following [26]; this second loss is given small weight which is reduced between epochs during training. Though we initialize θ_{Real} in the standard randomized way, the network is still encouraged to stay similar to the behavior of G_{Adobe} while having the flexibility to make significant changes that improve the quality of the mattes. We hypothesize that this formulation helps the network to avoid getting stuck in the local minimum of G_{Adobe}, instead finding a better minimum nearby for real data.

We use the LS-GAN [23] framework to train our generator G_{Real} and discriminator D. For the generator update we minimize:

$$\begin{aligned} \min_{\theta_{\text{Real}}} \mathbb{E}_{X, \bar{B} \sim p_{X, \bar{B}}} & [(D(\alpha F + (1-\alpha)\bar{B}) - 1)^2 \\ & + \lambda \{ 2\|\alpha - \tilde{\alpha}\|_1 + 4\|\nabla(\alpha) - \nabla(\tilde{\alpha})\|_1 \\ & + \|F - \tilde{F}\|_1 + \|I - \alpha F - (1-\alpha)B'\|_1 \}], \end{aligned} \tag{3}$$

where $(F, \alpha) = G(X; \theta_{\text{Real}})$, $\bar{B}$ is a given background for generating a composite seen by D, and we set λ to 0.05 and reduce by $1/2$ every two epochs during training to allow the discriminator to play a significant role. We use a higher weight on the alpha losses (relative to Equation 2), especially the gradient term to encourage sharpness.

For the discriminator, we minimize:

$$\begin{aligned} \min_{\theta_{\text{Disc}}} \mathbb{E}_{X, \bar{B} \sim p_{X, \bar{B}}} & [(D(\alpha F + (1-\alpha)\bar{B}))^2] \\ & + \mathbb{E}_{I \in p_{data}}[(D(I) - 1)^2], \end{aligned} \tag{4}$$

where θ_{Disc} represents the weights of the discriminator network and again $(F, \alpha) = G(X; \theta_{\text{Real}})$.

As a post-process, we threshold the matte at $\alpha > 0.05$, extract the largest N connected components, and set $\alpha = 0$ for pixels not in those components, where N is the number of disjoint person segmentations in the image.

Algorithm	Additional Inputs	SAD	MSE(10^{-2})
BM	Trimap-10, B	2.53	1.33
BM	Trimap-20, B	2.86	1.13
BM	Trimap-20, B'	4.02	2.26
CAM	Trimap-10	3.67	4.50
CAM	Trimap-20	4.72	4.49
IM	Trimap-10	1.92	1.16
IM	Trimap-20	2.36	1.10
Ours-Adobe	B	**1.72**	**0.97**
Ours-Adobe	B'	**1.73**	**0.99**

Table 1: Alpha matte error on Adobe Dataset (lower is better).

4. Experimental Evaluation

We compared our approach with a variety of alternative methods, esp. recent deep matting algorithms that have performed well on benchmarks: **BM**: Bayesian Matting [9] - traditional, trimap-based method that can accept a known background [8]. (An alternative, Poisson Matting [30, 12] with known background, performed much worse.). **CAM**: Context-Aware Matting [16] - trimap-based deep matting technique that predicts both alpha and foreground. **IM**: Index Matting [21] - trimap-based deep matting technique that predicts only alpha. **LFM**: Late Fusion Matting [36] - trimap-free deep matting algorithm that predicts only alpha.

4.1. Results on Synthetic-Composite Adobe Dataset

We train G_{Adobe} on 26.9k exemplars: 269 objects composited over 100 random backgrounds, plus perturbed versions of the backgrounds as input to the network. We train with batch-size 4, learning rate $1e^{-4}$ with Adam optimizer.

We compare results across 220 synthetic composites from the Adobe Dataset [35]: 11 held-out mattes of human subjects composed over 20 random backgrounds, in Table 1. We computed a trimap for each matte through a process of alpha matte thresholding and dilation as described in [35]. We dilated by 10 and 20 steps to generate two different trimaps (more steps gives wider unknown region). We additionally computed a perturbed background B' by applying small random affine transformation (translate $\in \mathcal{N}(0, 3)$, rotate $\in \mathcal{N}(0, 1.3°)$ and small scaling and shear) followed by gamma correction $\gamma \sim \mathcal{N}(1, 0.12)$ and gaussian noise $\eta \sim \mathcal{N}(\mu \in [-5, 5], \sigma \in [2, 4])$. For our approach, we only evaluated the result of applying the G_{Adobe} network ('Ours-Adobe'), since it was trained only on the Adobe data, as were the other learning-based approaches we compare to. We rescaled all images to 512×512 and measure the SAD and MSE error between the estimated and ground truth (GT) alpha mattes, supplying algorithms with the two different trimaps and with backgrounds B and B'

Figure 3: (a-e) Resulted alphas and foregrounds for photos taken with handheld camera against natural backgrounds; (e) is an example failure case with dynamic background (fountain). See video results in the supplementary.

as needed. We omitted LFM from this comparison, as the released model was trained on all of the Adobe data, including the test data used here (confirmed by the authors). That said, it produces a SAD and MSE of 2.00, $1.08e^{-2}$, resp., while our method achieves (true test) error of 1.72, $0.97e^{-2}$.

We observe that our approach is more robust to background perturbation when compared to BM, and it improves on all other trimap-based matting algorithms (BM, CAM, IM). As trimaps get tighter, the trimap-based matting algorithms get better, but tight trimaps are time-consuming to create in practice. The goal of our work is to fully eliminate the need for manually created trimaps.

4.2. Results on Real Data

We captured a mix of handheld and fixed-camera videos, taken indoors and outside using a smartphone (iPhone 8). The fixed-camera setup consisted of an inexpensive selfie stick tripod. In each case, we took a video with the subject moving around, plus a shot of the background (single video frame) with no subject. All frames were captured in HD (1920×1080), after which they were cropped to 512×512 (input resolution to our network) around the segmentation mask for one person or multiple. We retrain G_{Adobe} on 280k composites consisting of 280 objects from Adobe Dataset [35]. We then train separate copies of G_{Real}, one each on handheld videos and fixed camera videos, to allow the networks to focus better on the input style. For handheld videos we account for small camera shake by aligning the captured background to individuals frames through homography. In total, we trained on 18k frames for hand-held camera and 19k frames for fixed camera. We captured 3390 additional background frames for $\bar{B}$. We use a batch-size of 8, learning rate of $1e^{-4}$ for G_{Real} and $1e^{-5}$ for D and update D with Adam optimizer. We also update the weights of D after 5 successive updates of G_{Real}.

Ours vs.	much better	better	similar	worse	much worse
BM	52.9%	41.4%	5.7%	0%	0%
CAM	30.8%	42.5%	22.5%	4.2%	0%
IM	26.7%	55.0%	15.0%	2.5%	0.8%
LFM	72.0%	20.0%	4.0%	3.0%	1%

Table 2: User study on 10 real world videos (fixed camera).

Ours vs.	much better	better	similar	worse	much worse
BM	61.0%	31.0%	3.0%	4.0%	1.0%
CAM	43.3%	37.5%	5.0%	4.2%	10.0%
IM	33.3%	47.5%	5.9%	7.5%	5.8%
LFM	65.7%	27.1%	4.3%	0%	2.9%

Table 3: User study on 10 real world videos (handheld).

To compare algorithms on real data, we used 10 handheld videos and 10 fixed-camera videos as our (held-out) test data. The BM, CAM, and IM methods each require trimaps. We did not manually create trimaps (esp. for video sequences which is infeasible). Instead, we applied segmentation [4], and labeled each pixel with person-class probability > 0.95 as foreground, < 0.05 as background, and the rest as unknown. We tried alternative methods, including background subtraction, but they did not work as well.

To evaluate results, we could not compare numerically to ground truth mattes, as none were available for our data. Instead, we composited the mattes over a green background and performed a user study on the resulting videos. Since

IM and LFM do not estimate F (needed for compositing), we set $F = I$ for these methods. We also tried estimating F directly from the matting equation (given α and B'), but the results were worse (see supplementary material). We do not use any temporal information and set $M = \{I, I, I, I\}$ for all comparisons to prior methods.

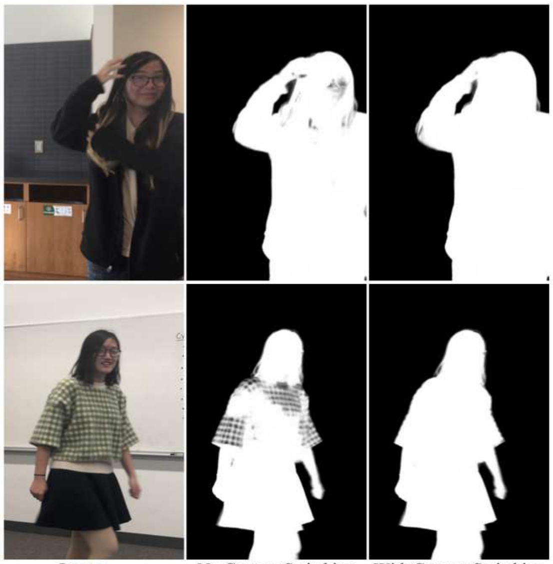

Figure 4: Role of Context Switching Block (CS Block).

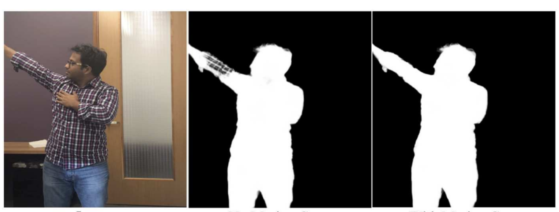

Figure 5: Role of motion cues.

In the user study, we compared the composite videos produced by G_{Real} network ('Ours-Real') head-to-head with each of the competing algorithms. Each user was presented with a web page showing the original video, our composite, and a competing composite; the order of the last two was random. The user was then asked to rate composite A relative to B on a scale of 1-5 (1 being 'much worse', 5 'much better'). Each video pair was rated $\sim$ 10 users.

The results of the user study, with scores aggregated over all test videos, are shown in Tables 2 and 3. Overall, our method significantly outperformed the alternatives. The gains of our method are somewhat higher for fixed-camera results; with handheld results, registration errors can still lead to matting errors due to, e.g., parallax in non-planar background scenes (see Fig 6(f)).

Single image results are shown in Figure 6, again demonstrating improvement of our method over alternatives. We note that LFM in particular has difficulty zeroing in on the person. More results generated by our approach with handheld camera in natural backgrounds are shown in Figure 3. In (c), (d) we show examples of multiple people interacting in a single image, and in (e) we show a failure case with a dynamic background, the fountain. **Please see supplementary material for video results and more image results.**

5. Ablation Studies

Role of motion cues. As shown in Figure 5, video motion cues M can help in predicting a cleaner matte when foreground color matches the background. (Note: we did not use motion cues when comparing to other methods, regardless of input source.)

	much better	better	similar	worse	much worse
handheld	16.4%	35.5%	42.7%	5.4%	0%
fixed-camera	17.3%	15.5%	51.8%	10%	5.4%

Table 4: User Study: Ours-Real vs Ours-Adobe.

'Ours-Real' vs 'Ours-Adobe'. As expected, 'Ours-Adobe' outperformed 'Ours-Real' on the synthetic-composite Adobe dataset on which 'Ours-Adobe' was trained. 'Ours-Real' achieved a SAD score of 3.50 in comparison to 1.73 of 'Ours-Adobe'. However 'Ours-Real' significantly outperformed 'Ours-Adobe' on real data as shown by qualitative examples in Figure 6 and by an additional user study (Table 4). The gain of 'Ours-Real' in the user study ($\sim$ 10 users per pair-wise comparison) was larger for handheld captures; we suspect this is because it was trained with examples having alignment errors. (We did try training 'Ours-Adobe' with alignment errors introduced into B' but found the results degraded overall.)

Role of Context Switching Block (CS Block). We compare our CS Block architecture to a standard residual-block-based encoder-decoder [38] scheme that was run on a naive concatenation of I, B', S, and M. We find that the concatenation-based network learns to focus too much on color difference between I and B' and generates holes when their colors are similar. The CS Block architecture effectively utilizes both segmentation and color difference cues, along with motion cues when present, to produce better matte, as shown in Figure 4 (more in supplementary). Empirically, we observe that the CS block helps significantly in 9 out of 50 real videos, especially when foreground color is similar to the background.

6. Conclusion

We have proposed a background matting technique that enables casual capture of high quality foreground+alpha mattes in natural settings. Our method requires the photographer to take a shot with a (human) subject and without,

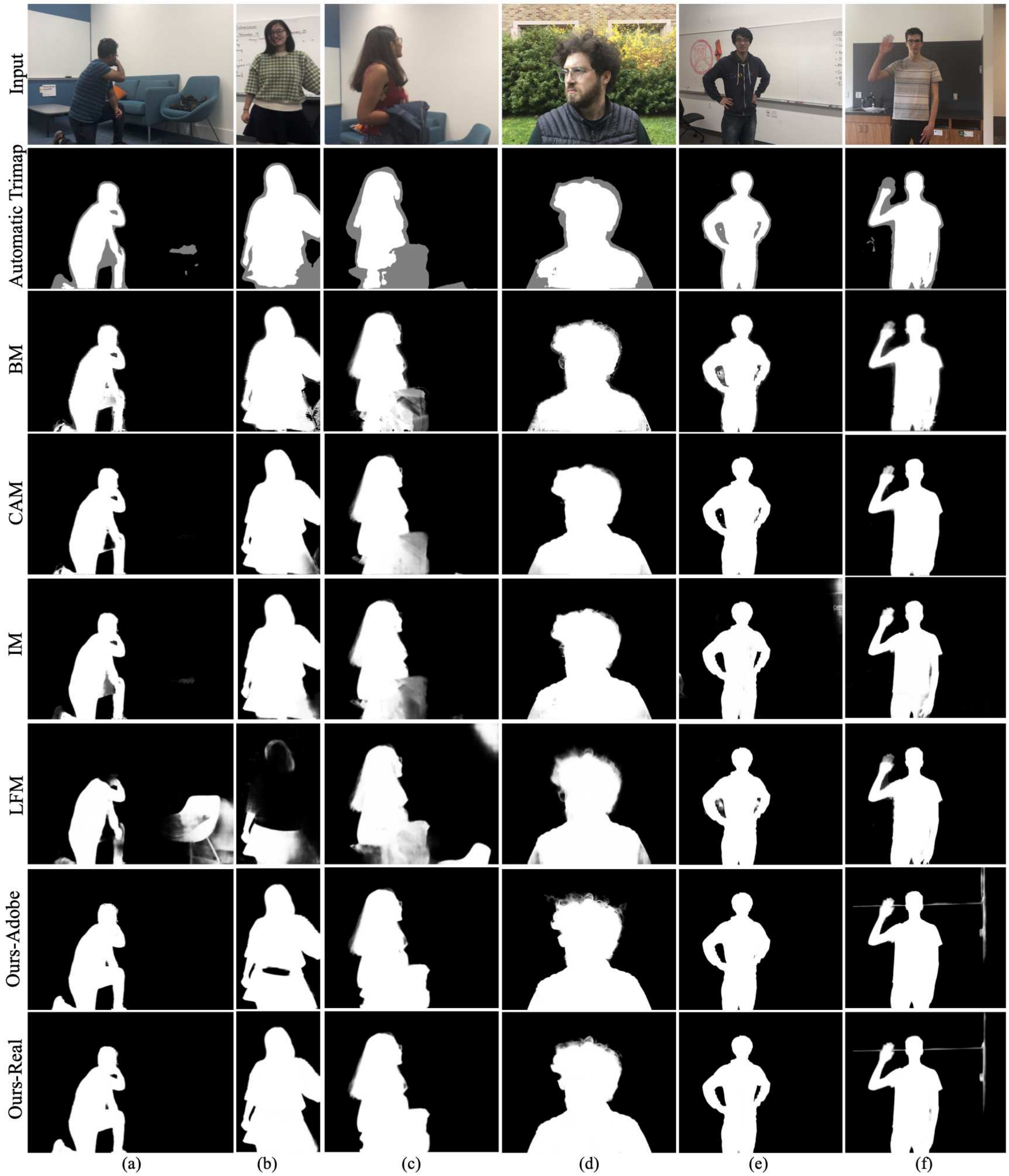

Figure 6: Comparison of matting methods with camera fixed (a,b,c) and handheld (d,e,f). Our method fails in (f) due to misregistration.

not moving much between shots. This approach avoids using a green screen or painstakingly constructing a detailed trimap as typically needed for high matting quality. A key challenge is the absence of real ground truth data for the background matting problem. We have developed a deep learning framework trained on synthetic-composite data and then adapted to real data using an adversarial network.

Acknowledgements. This work was supported by NSF/Intel Visual and Experimental Computing Award #1538618 and the UW Reality Lab.

References

[1] Yağiz Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4):72, 2018. 3

[2] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 29–37, 2017. 2

[3] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jiangyu Liu, Jiaming Liu, Jiaying Liu, Jue Wang, and Jian Sun. Disentangled image matting. *International Conference on Computer Vision (ICCV)*, 2019. 2

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 4, 6

[5] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. Semantic human matting. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 618–626. ACM, 2018. 3

[6] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2175–2188, 2013. 2

[7] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision*, pages 626–643. Springer, 2016. 2

[8] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H Salesin, and Richard Szeliski. Video matting of complex scenes. In *ACM Transactions on Graphics (ToG)*, volume 21, pages 243–248. ACM, 2002. 3, 5

[9] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *CVPR (2)*, pages 264–271, 2001. 2, 3, 5

[10] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer, 2000. 3

[11] Eduardo SL Gastal and Manuel M Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584. Wiley Online Library, 2010. 2

[12] Minglun Gong and Yee-Hong Yang. Near-real-time image matting with known background. In *2009 Canadian Conference on Computer and Robot Vision*, pages 81–87. IEEE, 2009. 3, 5

[13] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive alpha-matting. In *Proceedings of VIIP*, volume 2005, pages 423–429, 2005. 2

[14] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR 2011*, pages 2049–2056. IEEE, 2011. 2

[15] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2165–2172. IEEE, 2010. 2

[16] Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. *International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[17] Philip Lee and Ying Wu. Nonlocal matting. In *CVPR 2011*, pages 2193–2200. IEEE, 2011. 2

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

[19] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007. 2

[20] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1699–1712, 2008. 2

[21] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. *International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[22] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. *arXiv preprint arXiv:1807.10088*, 2018. 2

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017. 5

[24] Massimo Piccardi. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004. 3

[25] Richard J Qian and M Ibrahim Sezan. Video background replacement without a blue screen. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 4, pages 143–146. IEEE, 1999. 3

[26] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, refectance and illuminance of faces in the wild. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. 5

[27] Ehsan Shahrian, Brian Price, Scott Cohen, and Deepu Rajan. Temporally coherent and spatially accurate video matting. In *Computer Graphics Forum*, volume 33, pages 381–390. Wiley Online Library, 2014. 3

[28] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643, 2013. 2

[29] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *European Conference on Computer Vision*, pages 92–107. Springer, 2016. 2, 3

[30] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 315–321. ACM, 2004. 2, 3, 5

[31] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. Learning-based sampling for natural image matting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[32] Jue Wang and Michael F Cohen. An iterative optimization approach for unified image segmentation and matting. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 936–943. Citeseer, 2005. 2

[33] Jue Wang and Michael F Cohen. Optimized color sampling for robust matting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

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

[35] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2970–2979, 2017. 2, 3, 4, 5, 6

[36] Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. A late fusion cnn for digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7469–7478, 2019. 3, 4, 5

[37] Bingke Zhu, Yingying Chen, Jinqiao Wang, Si Liu, Bo Zhang, and Ming Tang. Fast deep matting for portrait animation on mobile phone. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 297–305. ACM, 2017. 3

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 4, 7