Color-aware Deep Temporal Backdrop Duplex Matting System

Hendrik Hachmann and Bodo Rosenhahn
Institute for Information Processing (tnt) / L3S - Leibniz University Hannover
Hannover, Germany
{hachmann,rosenhahn}@tnt.uni-hannover.de

ABSTRACT

Deep learning-based alpha matting showed tremendous improvements in recent years, yet, feature film production studios still rely on classical chroma keying including costly post-production steps. This perceived discrepancy can be explained by some missing links necessary for production which are currently not adequately addressed in the alpha matting community, in particular foreground color estimation or color spill compensation. We propose a neural network-based temporal multi-backdrop production system that combines beneficial features from chroma keying and alpha matting. Given two consecutive frames with different background colors, our one-encoder-dual-decoder network predicts foreground colors and alpha values using a patch-based overlap-blend approach. The system is able to handle imprecise backdrops, dynamic cameras, and dynamic foregrounds and has no restrictions on foreground colors. We compare our method to state-of-the-art algorithms using benchmark datasets and a video sequence captured by a demonstrator setup. We verify that a dual backdrop input is superior to the usually applied trimap-based approach. In addition, the proposed studio set is actor friendly, and produces high-quality, temporal consistent alpha and color estimations that include a superior color spill compensation.

CCS CONCEPTS

• **Computing methodologies** → **Video segmentation**; *Neural networks*; • **Human-centered computing** → *Virtual reality*.

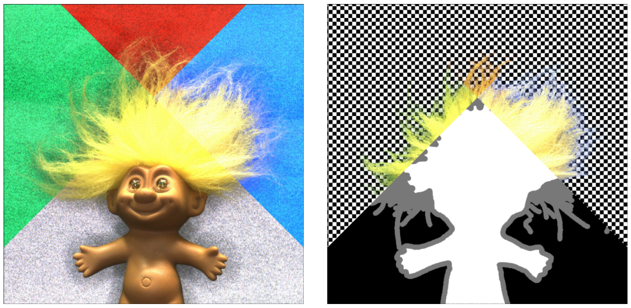
KEYWORDS

Alpha Matting, Color Spill, Neural Networks, Virtual Reality

ACM Reference Format:
Hendrik Hachmann and Bodo Rosenhahn. 2023. Color-aware Deep Temporal Backdrop Duplex Matting System. In *Proceedings of the 14th ACM Multimedia Systems Conference (MMSys '23), June 7–10, 2023, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3587819.3590973

1 INTRODUCTION

During the production of the 2019 Disney+ series "The Mandalorian", Industrial Light & Magic introduced StageCraft [1], a very high-definition LED video wall, in which visual effects are displayed on the wall, directly captured by the camera and thus appear in the footage. In this setup, traditional green screens are rendered unnecessary and no color keying or *pulling the matte* of objects or actors is needed, which simplifies the production and reduces visual effects (VFX) shot costs. Some say this may leave green screen technology obsolete. However, the option to change VFX at post-production is hereby abandoned. Nevertheless, the integration of similar video walls in studios may create interesting options for improved matting applications, one of which is proposed in this paper.



(a) Foreground

(b) RGBa and trimap

Figure 1: (a) a collage of 4 images of a troll in front of an retroreflective background, that has been illuminated with different colors. ((b), upper area) three compositions of an alpha-blended troll in front of a checkerboard background. The colored fringes are a problem called *color spill*, which is background color incorrectly extracted as foreground. ((b), lower area) a so-called *trimap*, that signals foreground, background and a gray area, the latter may contain transparencies.

Foreground transparency estimation is called *alpha matting*, with α being the amount of transparency for each pixel. $\alpha = 0$ denotes fully transparent foregrounds and $\alpha = 1$ meaning opacity. Formally, the matting equation

$$C = \alpha C_{fg} + (1 - \alpha) C_{bg} \tag{1}$$

needs to be solved, which is an ill-posed problem. The image C is often named *composition*, since it is an α-weighted superposition of the foreground color C_{fg} and the background color C_{bg}. Chroma keying refers to blue or green screen matting frequently applied in feature film production, a technique in which the color of the background is a priori given and thus key to the matting task. In production, the foreground color C_{fg} needs to be estimated as well as α, resulting in an RGBa stack.

A large number of green screen applications exist. They are used e.g. in news studios, weather forecasting, and film production. Chroma keying works well in expensive studios with highly controlled illumination. However, it poses algorithmic limitations as well as undesirable interference with actors.

MMSys '23, June 7–10, 2023, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0148-1/23/06.
https://doi.org/10.1145/3587819.3590973

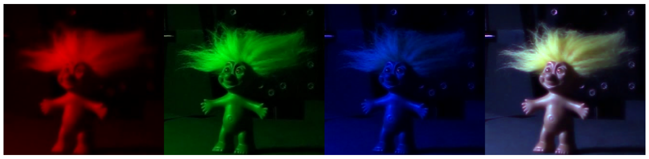


Figure 2: Images of a troll with yellow hair, that are captured with the same, static camera under varying monochromatic illuminations. It can be seen that the troll under green illumination appears detailed, while the red and blue illuminated images are blurry. This effect is called chromatic aberration, an effect caused by dispersion at the camera lens. The superposition of the three mono chromatic images is shown at the right. Here, the effect of chromatic aberration is visible as purple fringes.

There is obviously the limitation that the background color is to be avoided in the foreground. Furthermore, as can be seen in Figure 1, there is a problem called *color spill*: Visible background colored fringes shine through the foreground object at transparent points or are projected onto the foreground object from the side. Color spill is very noticeable in the composition with an alternative background and should be avoided or compensated for as much as possible.

In addition, matting is challenging if chromatic aberration [21] occurs. While this effect is often neglected in alpha matting literature, it is increasingly important if camera resolution increases or if consumer cameras are used. The effect of chromatic aberration is caused by dispersion at the camera lens. It can be seen in Figure 2, in which images of a troll are captured with monochromatic illumination. It can be observed that under green illumination the troll appears detailed, while the image becomes blurred or out of focus with red and blue illumination. This effect leads to colored fringes also known as rainbow edge in images with white illumination. In context of chroma keying systems using green or blue screens this effect can lead to a misperception as can be seen in Figure 3. The red hair of the troll is perceived less transparent in front of a green background compared to a blue background.

A color-aware matting system needs to predict foreground colors along with alpha mattes, while compensating color spill and being robust against chromatic aberration.

We summarise another category of problems as human discomfort impacting involved persons, i.e. newscasters or actors. Green screens must be illuminated very homogeneously. This is often achieved by strong illumination, which causes huge amounts of spotlights to heat up the room. In addition, humans feel the artificial environment, renders them disoriented and it is difficult for actors to put themselves into a scene, lacking so-called *immersive feedback*. That is why i.e. markers are often used so that actors at least look in the right direction of the VFX content. Markers, on the other hand, have to be masked to not influence chroma keying.

In this paper, we propose a temporally alternating backdrop matting system permitting dynamic cameras and foregrounds, alleviating foreground color restrictions, and allowing imprecise backings. The system deploys a one-encoder-dual-decoder neural network, that in an overlap-blend approach produces high-quality alpha and color estimation, including an advanced color spill compensation. The resulting simplification of studio sets along with high-quality matting can reduce production and post-processing costs. In addition, our system provides an actor-friendly environment with visual clues without any color restriction, enabling the actor to dive into the scene while performing.

The contributions of this paper are summarised by:

- We present a fully-functional temporal backdrop duplex setup, consisting of a camera and FPGA controlled LED panels synchronized @100 fps.
- Our hardware setup demonstrates the feasibility of an actor friendly studio set.
- A novel one-encoder-dual-decoder neural network architecture allows prediction of RGBa foregrounds from two consecutive frames with alternating backdrop color.
- The network handles dynamic scenes by combination of an inner patch prediction and an overlap-blend subdivision.
- We quantify the benefit of using dual backdrops instead of trimaps as input for neural alpha-matting.
- An automatic advanced color spill suppression method is proposed for post-production.

In the remainder of this paper, we review related work, describe our method and compare the performance to state-of-the-art approaches.

2 RELATED WORK

Studio sets: While the green or blue color is key for chroma keying, artifical homogeneous walls often leave actors without orientation. To regain orientation markers can be inserted that are later, sometimes even manually, masked out and removed from the footage. Tzidon and Tzidon [45] introduce a synchronized time duplex system in which markers are projected to a green screen at the readout time of the camera, generally called *blanking time*. Consequently, those markers do not appear in the footage at all. Vidal and Lafuente [47] use video projectors to add amplitudes of green to a green screen, without impacting the chroma keying. Within certain limits, these shades of green can give monochromatic visual clues to actors. Furthermore, Borja Vidal [46] uses polarized light and polarization filters to provide immersive feedback in combination with retroreflective screens. In this studio, the content is projected onto the background but is filtered so that the camera does not capture it. Grau *et al*. [11] use additional cameras to locate and track the actors' heads, in order to render VFX content based on the location of the actor. In this system, the actor sees view-dependent VFX content without geometric distortions to increase immersiveness into the scene.

Matting methods: Matting is a long-studied research field. In 2007, Wang and Cohen [49] published a survey paper on image and video matting, in which they describe and compare 50 matting approaches. The most common scenario is to predict the alpha matte given an input image and a trimap (see Figure 1), in which pure foreground and pure background are marked as well as a gray area, which is an unknown region that may contain transparencies. Trimaps are often seen as user input. For this matting task, there is an online benchmark by Rhemann[1] *et al*. [33], that is currently comparing 68 algorithms.

[1] alphamatting.com

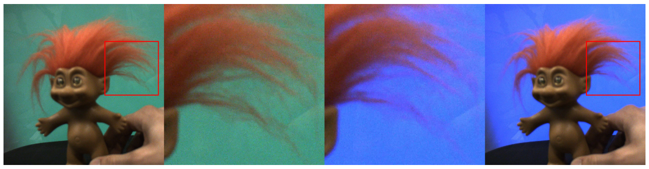


Figure 3: Effect of chromatic aberration illustrated by two consecutive frames taken form a troll sequence. The red hair is better visible in front of the green background compared to the blue background. Consequently, the hair strands on the left are perceived less transparent by humans and estimated by algorithms, which results in flickering alpha predictions in an alternating background color system.

To our knowledge, the first deep learning-based matting algorithm is *deep image matting* by Xu *et al*. [52]. They manually create a dataset with ground truth alpha mattes and train a fully convolutional neural network, that given a stack of RGB images and a trimap predicts an alpha matte. The network consists of an encoder-decoder stage predicting a coarse alpha matte and a refinement stage that locally improves the results. Since then researchers argue that the natural structure of foregrounds is inherently learned by neural networks which provides superior performance compared to traditional alpha matting. Recent approaches such as Sun *et al*. [41] combine the matting with a classification task. This semantic image matting uses multiple object categories and individual matting networks are trained. Given hardware memory limitations matting on high-resolution images becomes challenging, which is why Yu *et al*. [53] introduce a patch-based method in which query patches are compared to context patches from different image regions to increase intra-image consistency. Similarly, consistency across scales or hierarchical structures are optimized [25, 31]. Animal matting is particularly challenging due to fur and camouflage effects. Thus Li *et al*. [24] introduce a matting network in which two separate glance and focus networks work together combining the task of recognizing animals and locally extracting fur details.

In feature film production, a trimap is not given. Instead, the color key separates the foreground from the background. A coarse foreground estimation can be generated by different means: Recent progress in face detection initiated a sequence of portrait matting publications [19, 20, 23, 37], background subtraction is adopted by Sengupta *et al*. [35], saliency maps by Gupta and Raman [13], and attention by Zhou *et al*. [56] and Zhang *et al*. [54].

Alpha matting can also be used on videos. Erofeev *et al*. [8] maintain a video matting benchmark[2] created by triangulation in a stop motion fashion. In video matting, typically information is propagated from one frame to the next. For this task, rotoscoping can be applied which is the tracing of shapes or foregrounds in a sequence of images. Agarwala *et al*. [2] reduce the manual work of a human in the loop by semi-automatic rotoscoping. Today, this process is automated [4, 5, 30] and temporal consistency enforced [22, 27, 36, 42, 54].

Many matting approaches predict mattes only. However, for most applications this represents just one of two parts, since the foreground colors need to be estimated as well. Occurring color spill, as a result of imprecise foreground colors is often seen as an independent problem. FBA matting by Forte *et al*. [10], SIM by Sun *et al*. [41] and the method of Hou and Liu [15] are neural networks that simultaneously predict alpha and foreground colors. FBA is currently the leading algorithm on the benchmark of Erofeev *et al*. [8].

The effect of color spill (see Figure 1) for blue screen keying and a compensation technique was published as early as 1977 by Petro Vlahos [48]. Since then, the problem has not been fully solved, especially for non-perfect backing information. A common concealment practice is to just reduce the saturation of the foreground color in transparent areas, since a gray color spill attracts less attention. In feature film production color spill removal still requires manual work. More recently, Teng *et al*. [43] introduce a matting method for non-uniform illuminated blue screens.

As a note, convolutional neural networks have successfully been applied to matching and optical flow estimation as in FlowNet [9], FlowNet 2.0 [16] and deep convolutional matching [17, 32]. All of them can jointly process information that is spatially separated.

According to the survey of Wang and Cohen [49] our matting approach would be classified as “matting with extra information”. Those matting algorithms e.g. use flash image pairs as Sun *et al*. [40], camera arrays as Neel *et al*. [18], defocussing as McGuire *et al*. [29] or passive polarization as McGuire *et al*. [28]. Others use, as we do, multiple backgrounds with changing colors as Smith and Blinn[39] and Grundhöfer *et al*. [12]. These two methods share a common hardware setup with our approach. Therefore, the next paragraph presents them in detail and they are included in our experimental evaluation (cf. Section 4).

Smith and Blinn [39] propose a system directly linked to the matting equation 1, which becomes overdetermined and thus solvable for static scenes if two known backgrounds are used. The corresponding method is called *triangulation*. For each pixel it can be implemented as a system of linear equations and is frequently used to generate ground truth datasets. Triangulation can be used to exactly calculate alpha values and foreground colors. The method is also applied by Erofeev *et al*. [8] and by Rhemann[3] *et al*. [33].

Another multi-background matting system is proposed by Grundhöfer *et al*. [12], in which they are chroma keying video frames with alternating complementary background colors. In offline mode, trimaps are created by the color difference of backgrounds and Bayesian matting [6] is applied. For static scenes, similar to Smith and Blinn [39] this system can create perceptual high-quality mattes by superposition of two mattes since the color spill in both backdrop colors adds up to a neutral “white”. Being aware that foreground movements introduce errors they apply a seam color compensation heuristic that conceals errors.

While these temporal backdrop systems are similar in hardware, our deep learning-based matting technique is capable of handling moving foregrounds and backgrounds, is not restricted to precise knowledge of backing colors, and is superior in color spill compensation. Internally our proposed matting system can partly be seen

[2] videomatting.com

[3] alphamatting.com

as a frame-wise registration system, followed by matting of registered foregrounds with two known backings which can then be done flawlessly.

3 METHOD

3.1 Time duplex system

As illustrated in Figure 4, our proposed studio hardware setup consists of a freely moving global shutter camera (a FLIR ORX-10g-51S5-C set to a resolution of 2448×1600 @100 fps), a diffusor and a synchronized RGB LED wall. The diffuser smoothes individual LED spots of the 6 mm pitch modules so that individual LEDs are not recognizable in the camera image. Our demonstrator consists of 4 panels each with 32×32 individually controllable LEDs. The panels are aligned to a roughly 50 cm by 50 cm "wall", which, however, can easily be increased to any desired size. The panels are controlled by a BeagleBone Black and a LogiBone with a Xilinx FPGA according to a description by Glen Akins [3]. A signal generator synchronizes the camera and the BeagleBone. The LED wall displays the following sequence: a homogeneous green screen, VFX content, and a homogeneous blue screen followed by VFX content. Then the cycle starts again from the beginning. The system is synchronized so that the camera only captures the green and blue screen during the 1 ms exposure time respectively, but not the VFX content, which is shown for 9 ms during the camera's blanking time. The human eye, on the other hand, interpolates the full 10 ms and therefore mainly perceives the VFX content. Using our demonstrator, we simulate VFX content by a homogeneous red background, which at 100 fps is visible to us without any flickering on our wall. In detail, our panel illuminates only four of the 32 LED rows simultaneously, while the other 28 rows are switched off, which is called a 1:8 scan rate. Our FPGA implementation ensures that all rows are turned on for the same amount of time within the 1 ms global shutter exposure time, which is why we can achieve a homogeneous illumination across the whole panel. To sum up, our system can provide chroma keying information to the camera as well as different information for actors, e.g. VFX content or markers.

3.2 Multiple backdrop matting

Dataset: Our image processing pipeline (Figure 4) is built on a deep neural network that predicts alpha mattes based on incoming camera frames. This network is trained on the Adobe dataset [52], which provides 432 foreground images with corresponding ground truth alpha mattes. In order to adapt to our use-case, we create our own background dataset, which either includes pure green ($RGB = [0, 255, 0]$) and pure purple ($RGB = [255, 0, 255]$) backgrounds or consists of two consecutive frames taken from a 5000 frame video. This video is acquired with our demonstrator that captures the LED panel wall from different angles, resulting in green and blue backings as can be seen in Figure 6, 8 and 10. These real-world backgrounds have challenging properties: their color varies i.e. because of unwanted reflections, changing viewpoints, and noise from the camera's image sensor. These fore- and backgrounds are strictly divided into training and validation samples, with a split of 80% to 20%. Random selection of foreground-background combinations creates a 34560 samples training set and a 8640 samples validation set.

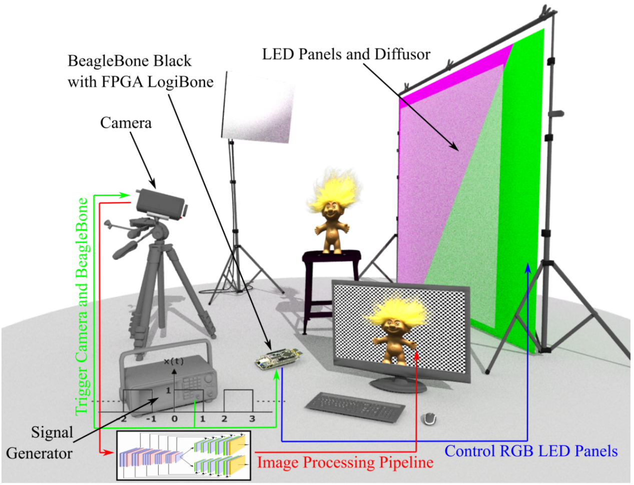


Figure 4: Studio setup: a camera captures a person or an object in front of an LED panel wall with a diffusor, which can provide chroma keying information, e.g. a green or blue screen for matting. In time duplex VFX content or markers for actors are displayed on the same wall as well. The panels are controlled by a BeagleBone and the system is synchronized by a signal generator. For each frame, the image processing pipeline creates a composition of the extracted foreground and a new background.

Deep neural network: The focus of this work is on the overall matting system rather than a highly optimized network architecture. Thus, our network structure (Figure 5) is by design very traditional and close to Xu *et al.*'s [52] encoder-decoder stage. The input to our network are two patches p_1 and p_2, which are extracted from consecutive RGB frames (f_1 and f_2) with different background colors. The decoder is realized by two separate branches that each outputs 4-channels consisting of RGB foreground color and an alpha estimation. We add the second decoder branch because our network processes patches from consecutive frames f_1 and f_2. The encoder and decoder are identical in construction to Xu *et al.* [52], using 14 convolutional layers with ReLUs and 5 max-pooling layers. Each of the two decoders has 6 convolutional layers with ReLUs and 5 unpooling layers. We add skip connections between feature maps of encoder and decoder.

Training: At the training of our network, both decoder outputs are compared to the corresponding alpha and RGB ground truth. The loss is defined as

$$\begin{aligned}
\mathcal{L}_\alpha^{i,j} &= \sqrt{(\alpha_p^{i,j} - \alpha_g^{i,j})^2 + \epsilon^2}), \\
\mathcal{L}_c^{i,j} &= \sqrt{(c_p^{i,j} - c_g^{i,j})^2 + \epsilon^2}), \\
\mathcal{L}^i &= \omega_\alpha \mathcal{L}_\alpha^{i,1} + \omega_\alpha \mathcal{L}_\alpha^{i,2} \\
&\quad + \omega_c \mathfrak{m}^{i,1} \mathcal{L}_c^{i,1} + \omega_c \mathfrak{m}^{i,2} \mathcal{L}_c^{i,2}
\end{aligned} \tag{2}$$

in which $\mathcal{L}^i$ is the (ω_α,ω_c)-weighted superposition of the α-prediction loss $\mathcal{L}_\alpha^{i,j}$ and the color prediction loss $\mathcal{L}_c^{i,j}$, with i indicating pixels and $j \in 1, 2$ frames. The α-prediction loss $\mathcal{L}_\alpha^{i,j}$ measures the α-value prediction $\alpha_p^{i,j}$ in comparison to the ground truth $\alpha_g^{i,j}$. Similar, the color loss $\mathcal{L}_c^{i,j}$ is calculated comparing the predicted color $c_p^{i,j}$ for each channel to the ground truth $c_g^{i,j}$. All 4 channels are scaled to a range of $[0, 1]$. The color loss is $\mathcal{L}_c^{i,j}$ masked and only active for pixels where

$$\mathfrak{m}^{i,j} = \begin{cases} 0, & \text{if } \alpha_g^{i,j} = 0, \\ 1, & \text{if } \alpha_g^{i,j} > 0 \end{cases} \tag{3}$$

since we do not want the network to estimate foreground colors if the foreground cannot be seen. Similarly, in case of foreground movements, prediction of parts that are not visible in both input patches is prevented by limiting the output region. Thus, the overall loss $\mathcal{L}_{overall} = \sum_{i \in \gamma} \mathcal{L}^i$ is evaluated at the inner region γ of each patch only, leaving a loss-free region of 50 pixels surrounding γ (see Figure 6 and 7). In doing so, we ensure that each inner region pixel exists in both input frames, even if the foreground moves by up to 50 pixels.

As part of our training, we use foreground and background displacement augmentation (Figure 7), simulating foreground movements and a freely moving camera. In detail, given two consecutive frames of our background dataset, we first sample a position A within the background, composite a foreground, and cut out the first training input by sampling a position B. Then, we randomly sample two vectors $V_{Foreground}$ and V_{Cutout} that displace foreground and cutout positions for the second frame f_2 and cut out a patch p_2. As before, we limit all movements to a maximum of 50 pixels.

We follow a similar approach as Xu *et al.* [52] and crop image pairs to different sizes (320×320, 480×480, and 640×640) and downscale them to patches of size 320×320, thus covering multiple scales. Further training details can be found in Section 3.3.

Network deployment: As part of our image processing pipeline, we calculate foreground and alpha estimations for the consecutive frames f_1 and f_2. In the remainder of this section, we describe the network deployment or inference for the first frame f_1 only, which is similarly applied for the second frame f_2. We define $f_1 = f$, $p_1 = p$ and $\gamma_1 = \gamma$. Similar to Yu *et al.* [53], the incoming frame f is subdivided into a set of overlapping patches $p^a, p^b, p^c, ... \in P$, which are sequentially processed by the network and we gain a set of overlapping predictions, the corresponding patches $\gamma^a, \gamma^b, \gamma^c, ... \in \Gamma$. All patches in the set P are of size 320×320 and all inner patches in Γ are of size 220×220. The overlap between neighboring patches in P is 100 pixels so that also the inner patches Γ overlap by 50 pixels. Two neighboring patches $\gamma^a, \gamma^b \in \Gamma$ are linearly blended in the overlapping area $\gamma^a \cup \gamma^b$, with blending weights proportional to the distances to the patch boundaries (see [53]). Thus the influence

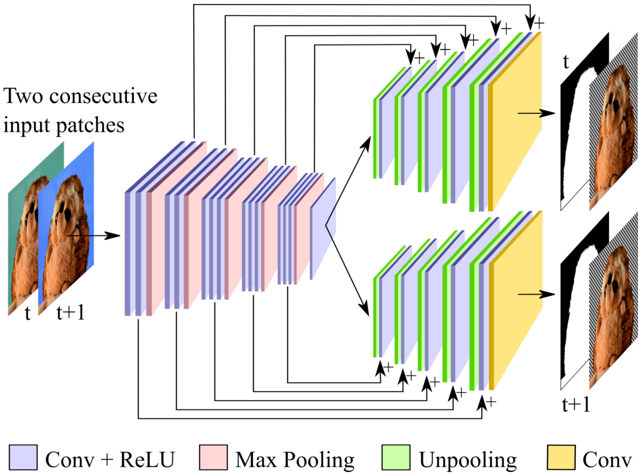


Figure 5: Given two sequential input patches, our convolutional neural network predicts an alpha matte as well as foreground colors for each frame, using a single-encoder-dual-decoder architecture with skip connections.

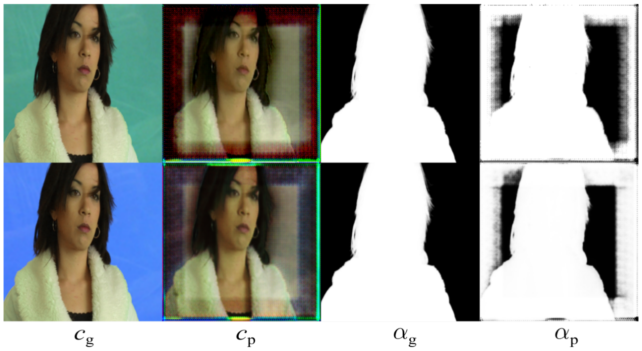


Figure 6: Comparison of foreground color c_p and alpha α_p predictions and the ground truth (c_g, α_g). The loss of our neural network is only active at the center of each patch p, which we call region γ, leaving a 50 pixel "don't care"-area at the boundaries. It is very important to note, that both color predictions c_p (top and bottom) do not contain green or blue color spill in the inner patches.

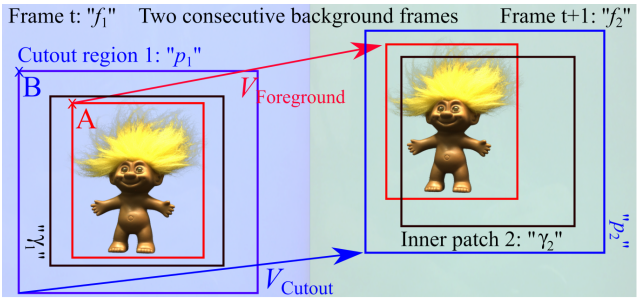


Figure 7: We operate on different image sizes. Consecutive camera frames are denoted as f_1 and f_2. Our neural network operates on patches p_1 and p_2 and on inner patch regions γ_1 and γ_2. In order to train our network, we augment two input patches with simulated foreground and background movements. Given two background frames, we randomly sample a foreground position A, a cutout position B and composite the first input patch p_1. We displace foreground and cutout region by two randomly sampled displacement vectors $V_{Foreground}$ and V_{Cutout} for the second input patch p_2. Similarly, we obtain our ground truth alpha matte.

of the prediction γ^a gradually diminishes at its boundaries with the increasing influence of the neighboring prediction γ^b.

Following this overlap-blend approach, we obtain full-size foreground predictions with colors $C_{fg,pred}$, and alphas α_{pred} for each frame f. For the composition of the foreground prediction with a new background C_{bg}, i.e. VFX content, we modify the original matting equation 1 to

$$\begin{aligned} C = \alpha_{pred}(\alpha_{pred}C_{fg,pred} + (1 - \alpha_{pred})C_{fg,orig}) \\ + (1 - \alpha_{pred})C_{bg}. \end{aligned} \tag{4}$$

Thus, we blend the color information, using the original colors $C_{fg,orig} = f$ if α_{pred} is close to 1 and the predicted color $C_{fg,pred}$ if α_{pred} is close to 0. In short, equation 4 introduces our color spill correction.

3.3 Implementation details

In the following, we want to add further details on training and implementation of our method and our implementation of DIM [52] to facilitate reimplementations. These details are described detached from the description of the method in order to increase readability of the previous subsection.

Architecture: The architecture of our network can be seen in Figure 5. Input dimensions are 6×320×320 and output dimensions are 4×320×320, for each decoder. The encoder is similar to Xu *et al.* [52] and consists of blocks with 2 or 3 2D convolutional layers (kernel size 3), followed by group normalization (as introduced by Wu and He [51]), rectified linear unit (ReLU) activation and max pooling. With each encoder block, the output shape is reduced by a factor of 2, while the number of channels increases in the sequence 64, 128, 256, 512 to 1024. The encoder parameters of the 14 layers are initialized by the pre-trained VGG16 network of Simonyan and Zisserman [38]. Both decoders have the same architecture with blocks consisting of convolutional layers followed by group normalizations, ReLUs, and 2D transposed convolutions (kernel size 6 and stride 2). The final convolution (yellow in Figure 5) is followed by a ReLU activation and clipping of values to a maximum of 1. This is motivated by the observation that color values tend to be evenly distributed. This is in contrast to alphas values, which are typically close to one or close to zero, for which sigmoid activation is preferred. The decoder weights are initialized randomly. Similar to Forte and Pitié [10], we use a mini-batch size of 1. The long skip connections (Figure 5), connecting encoder and decoder, are motivated by U-nets introduced by Ronneberger *et al.* [34] and which are beneficial for the prediction of fine-grained details. In our implementation, the skip connections are additive as in ResNets introduced by He *et al.* [14]. Our dual decoder network consists of 81.59M trainable parameters.

Parameters: In Equation 2, ω_c is set to 0.5 and ω_α to 1. As described in the paper, the color loss $\mathcal{L}_c^{i,j}$ can be masked by $\mathfrak{m}^{i,j}$. If it is not masked, meaning $\mathfrak{m}^{i,j} \neq 0$, color prediction errors have a larger impact on the loss $\mathcal{L}^i$ than alpha prediction errors, since three channels are each weighted by $\omega_c = 0.5$. In Equation 2, ϵ is set to $1e - 6$. In Figure 7, each displacement vector $V_{Foreground}$ and $V_{Background}$ is limited to 50 pixels, resulting in a maximal combined foreground-background movement of 100 pixels.

Training: As training dataset we use the 432 foreground images of the Adobe dataset [52] and 5000 frames of recorded background samples from our demonstrator setup. Foreground and background samples are divided into training (80%) and validation (20%) data each. Within each dataset (training and validation) foreground and background combinations are sampled resulting in a training set of 34560 and a validation set of 8640 foreground-background tuples. The data split and the combinations were created once and stayed the same during training. A tuple consists of one foreground image and two consecutive background images with different background colors. During training, within one foreground-background tuple, random cropping augmentation is conducted as illustrated in Figure 7. This randomized real-time augmentation leads to increased data diversity. In addition, augmentation techniques such as randomly flipping, changing contrasts, or adding color jitter are applied. Our network is implemented in pytorch. As optimizer we chose stochastic gradient descent with an initial learning rate of 0.01 and momentum of 0.9. We enforce a steadily falling validation loss by reinitialization with a saved checkpoint from the previous epoch, if the validation loss increase. If no decrease in validation loss is achieved within 2 epochs, the learning rate is decreased by a factor of 0.6. If no decrease in validation loss is achieved within 5 epochs, training is terminated. The network was trained for 50 epochs, which took 89 hours using an Nvidia GeForce RTX 2080 Ti GPU.

Inference: During training, we apply a randomized movement augmentation to simulate foreground and background movements, to increase the robustness of the network for moving scenes. During inference, meaning the deployment of the network as part of the proposed matting system, overlapping patches are cropped from the camera frames f_1 and f_2 in a fixed grid and corresponding patches p_1 and p_2 are extracted from the same coordinates but from two consecutive frames, meaning that using a camera speed of @100 fps the second frame f_2 was captured 10 ms after f_1. Thus, moving or non-static foregrounds and backgrounds have a displacement in p_1 and p_2.

4 RESULTS

Algorithms: We compare our method to the following related algorithms. The first method is denoted *BSM*, which is our implementation of Smith and Blinn's [39] blue screen matting. The authors of BSM claim that matting with a "multi-background technique" can only be applied in a static case, without "live actors or other moving objects". In addition BSM needs a perfect knowledge of the background colors. Furthermore, Smith and Blinn write that an application without these requirements is "powerful". We see our approach as an extension of their work, which is able to overcome these limitations and show that multi-background matting can be applied on dynamic scenes.

The second method *CIM* is our implementation of Grundhöfer *et al.*'s [12] method, in which they use hardware keying units (*Ultimatte 11*) with additional Bayesian matting. Since we do not own Ultimatte 11, we cannot recreate this particular matting pipeline. Instead, CIM uses ground truth alpha mattes or "learning-based digital matting" by Zheng and Kambhamettu [55] on the *Troll* sequence, for which no ground truth is available. CIM requires complementary backing colors, that sum up to a neutral white, which is the core idea of

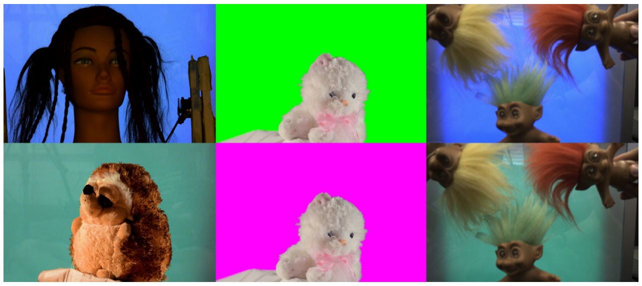


Figure 8: Evaluation dataset: (left) Composition of *Castle* and *Dmitriy* of Erofeev *et al.*'s [8] benchmark[5] in front of a realistic blue and green background respectively, (middle) composition of *Alex* (also [8]) in front of a pure and orthogonal green and purple background and (right) two consecutive frames with *Trolls* using our demonstrator.

their color spill neutralization. Therefore, similar to BSM, perfect backgrounds are needed, however, this requirement is only fulfilled by the pure color background sequences.

In contrast to the previous two alternating backdrop methods, the following two methods are trimap-based. In our scenario trimaps are generated from the ground truth alpha matte by setting all pixels that fulfill $0 < \alpha < 1$ to "unknown", followed by morphological dilation of this gray zone. The distribution of foreground, background and unknown areas of the test sequences can be seen in Table 1.

The third matting technique *DIM* is an implementation of Xu *et al.* [52]. The basic structure of this neural network can be explained with the help of Figure 5, applying a few modifications. Instead of the second input patch p_2, the network receives a trimap and the second decoder branch is omitted. By comparing to DIM we can directly measure any performance gain achieved by substituting the trimap version with our dual-frame version.

The forth method Semantic Image Matting or *SIM* by Sun *et al.* [41], we use official repository, enhances the matting results by first classifying the foreground, creating a semantic trimap which guides the matting network. In our evaluation the classification into the classes *fur*, *hair_hard* and *hair_easy* should be beneficial along with *motion* for the Trolls sequence. The class *defocus* may help to cope with chromatic aberration. SIM ranks in the top-5 of Rhemann[4] *et al.* [33] benchmark.

Finally, we compare with *FBA*, the official demonstrator of Forte *et al.* [10], which is publicly available. The method is currently the leading algorithm on the benchmark[5] of Erofeev *et al.* [8] and represents the state-of-the-art in trimap-based matting.

Test dataset and metrics: We evaluate our method on the three sequences from Erofeev *et al.* [8], called *Dmitriy*, *Alex* and *Castle*. While this dataset includes more sequences, these are the only sequences for which ground truth alpha mattes are publicly available. In this paper, these are necessary for the alternating color background sequence generation and for evaluation. The foregrounds Dmitriy, Alex and Castle are composited with alternating blue and green backdrops acquired by our demonstrator or with complementary pure green and pure purple backings. The resulting samples can be seen in Figure 8 and the top row of Figure 10. Evaluation on composition level has several advantages compared to independent evaluation of alpha values and foreground color prediction, since the effect of errors in foreground color prediction are linked to alpha values. If alpha is zero, colors may be erroneous without impacting the composition. Evaluation on the composition solves this issue by measuring alpha and color estimation at the same time. Furthermore, some measures, such as Gradient and Connectivity (see below), do not make sense for colors.

	Alex	Dmitriy	Castle	Troll
White area in trimap	18.35%	23.81%	29.66%	29.87%
Black area in trimap	71.35%	65.75%	23.50%	31.13%
Gray area in trimap	10.29%	10.44%	46.84%	39.01%
max(PSNR)	60.0	60.0	60.0	n.a.
max(VMAF)	99.86	99.98	97.43	n.a.
max(MS-SSIM)	0.9999	0.9999	0.9999	n.a.

Table 1: Dataset properties. Top rows: trimap color distribution with the percentage of foreground (white pixel), background (black pixel) and unknown region (gray pixel). Bottom rows: upper metric limits for PSNR, MS-SSIM and VMAF.

For quantitative evaluation, the matting results are composited with checkerboard backgrounds, compared to the ground truth, and evaluated with the following metrics: PSNR, MS-SSIM [50] and the perceptual Video Multi-Method Assessment Fusion (VMAF) [26]. The peak-signal-to-noise ratio (PSNR) is based upon the mean squared error (MSE), which measures a pixel-wise comparison to the ground truth. Temporal inconsistencies or flickering are of major importance, since the human visual system is strongly affected by them. Furthermore, error concealment methods seldomly correct matting results, but diminish the impact of occurring errors. This is why we also evaluate the structural similarity (MS-SSIM) and perceptual (VMAF) scores. We calculate all metrics using FFmpeg [44] and lossless H.264 encoding ($qp = 0$) with $YCbCr = 4{:}4{:}4$, meaning without chroma subsampling. The highest achievable scores can be obtained by comparing the ground truth to itself. These upper metric limits can be found in Table 1 and it can be seen that the maximal VMAF score is dataset dependent. The metrics PSNR, MS-SSIM and VMAF are frequently used in the video coding community that has a long record of in-depth video quality assessment.

In the matting community, the alpha mattes are typically evaluated independently from the foreground colors. In Table 2, we show results on the alpha prediction measures SAD, MSE, Gradient and Connectivity as proposed by Rhemann *et al.* [33]. The results of our method ours$_{ma}$ on these scores are on a similar level as FBA and SIM and far better than DIM, BSM and CIM. Nevertheless, this paper focuses on foreground color prediction as a part of a matting system and thus the results on the scores MS-SSIM, PSNR and VMAF are of major importance.

As part of our assessment, we provide quantitative results on our *Troll* sequence, a video captured by our demonstrator. In contrast to the virtual dataset, the Troll dataset has additional challenges

[4] alphamatting.com
[5] videomatting.com

such as noise, chromatic aberration, and non-homogeneous, not fully known backgrounds. The *Troll* dataset can only be used for qualitative evaluation as no ground truth exists. The Troll dataset and other sequences from the demonstrator are publicly available at (anonymous_submission).

Discussion: Quantitative results of seven algorithms on our test dataset with real-world backgrounds can be found in Table 2 and the following five key findings can be observed.

First, our method ours$_{ma}$ performs better than BSM [39] and CIM [12] by a large margin. This comparison is important as these three methods receive the same input data and can share the same hardware setup.

Second, the comparison of DIM [52] to our method shows the direct benefit of replacing the trimap with a 2 frame input, since both networks are otherwise similar in architecture and trained on the same dataset with identical hyper parameters. The results show a drastic gain in performance, which is directly linked to the architecture changes introduced with our method.

Third, our method performs on a similar level as FBA by Forte *et al.* [10] and SIM by Sun *et al.* [41], which proofs that ours$_{ma}$ can provide state-of-the-art results. Note that FBA and SIM need a trimap as input which is not easily obtained and any errors in the foreground and background areas of the trimap directly lead to errors in the alpha mattes. In our evalutation, the trimaps given to FBA and SIM are without errors and contain 53.16% to 89.71% of ground truth data (white and black area in Table 1).

Fourth, an ablation study illustrates the impact of our motion augmentation. The column ours$_{sa}$ shows results of our network trained statically, while ours$_{ma}$ used motion augmentation during training. Static means that $V_{Foreground} = 0$ and $V_{Cutout} = 0$ (cf. Figure 7). From all scores, it can be observed that the motion augmentation is of enormous importance. In Figure 9 both models show qualitative results on two frames of the non-static Castle dataset. The static model ours$_{sa}$ clearly fails to cope with foreground movements, which leads to double contours in the alpha matte and increased color spill in the composition with a checkerboard background.

Fifth, Table 2 provides MAD scores on the Troll sequence, for which no ground truth alpha values are available. However, we can measure temporal alpha value consistency as follows. With changing background colors the average alpha values of the foreground should remain unchanged. The accumulated mean alpha value deviation

$$MAD = \frac{1}{N} \sum_{n=1}^{N-1} \frac{1}{wh} \left| \sum_{x=1}^{w} \sum_{y=1}^{h} \alpha_n(x,y) - \sum_{x=1}^{w} \sum_{y=1}^{h} \alpha_{n+1}(x,y) \right| \quad (5)$$

measures alpha estimation inconsistencies between frames on the Troll dataset, with $N = 25$ the number of frames, alpha values ranging from 0 to 255 and a video resolution with width $w = 2448$ and height $h = 1600$. In theory, MAD may also consist of an alpha deviation due to foreground movements and deformations. However, this share of the MAD score is the same for all algorithms. Table 2 illustrates that the alpha predictions of ours$_{ma}$ achieve the best MAD score and are thus most consistent.

As a sanity check we evaluate on an artificial dataset with pure green and purple backgrounds (see center columns in Figure 8). As we expect, BSM performs best and creates perfect results on Dmitriy and Alex (see Table 1). Only the Castle sequence scores are non-perfect with MS-SSIM 0.9993, PSNR 49.90 and VMAF 97.20, which is probably due to numerical inaccuracies.

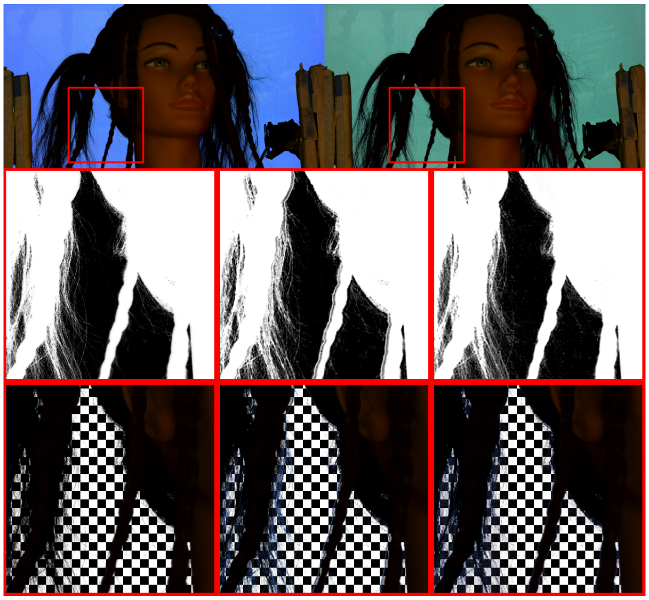


Figure 9: Effect of motion augmentation illustrated on the Castle sequence: (top): input frames to the network and overview of the cropped region, (lower, left): ground truth alpha matte and ground truth superposition with a checkerboard background corresponding to the blue background input image (lower, center): predictions by the proposed network ours$_{sa}$ trained without motion augmentation or static and (lower, right): predictions by the proposed network ours$_{ma}$ trained using motion augmentation. The lack of motion augmentation during training leads to double contours and increases color spill.

Qualitative results on our test dataset with real-world backgrounds and the Troll sequence can be found in Figure 10. In the first three columns, it can be seen that the foreground movement has a drastic impact on BSM and CIM, which show color seams and erroneous transparencies in the foreground. It can be seen on the lower row of CIM's Castle examples that superimposing alpha mattes without registration leads to visible double contours at single hair strands. On the right column, in the Troll sequence, we observe that DIM, FBA, SIM and CIM suffer from blue color spill, while our method shows close to no color spill, a slight blue shade in the green hair merely. Quite curiously, BSM seems to have shades of red in the yellow hairs and yellow shades in the green hairs, which seems to be the triangulation error. Furthermore, FBA, SIM and CIM seem to be affected by the noise in this dataset.

A comparison of the consecutive frames from the Troll dataset, as in Figure 3, 8 and 10, reveals that single hair fibers can be more easily recognized with green backing, which is most probably due to chromatic aberration [21]. As can be seen in Figure 3, this effect leads to overestimated alpha values for green and underestimated alpha values for blue backdrops, which we experience systematically with all seven matting algorithms. While the effect of chromatic

Data	Metric	better	DIM[6] [52]	FBA [10]	SIM [41]	BSM[6] [39]	CIM[6] [12]	ours$_{sa}$	ours$_{ma}$
Trimap			✓	✓	✓	-	-	-	-
2 frames			-	-	-	✓	✓	✓	✓
Alex	MS-SSIM	↑	0.98332	**0.99691**	0.99544	0.93664	0.96418	0.95779	**0.99714**
Dmitriy	MS-SSIM	↑	0.98984	0.99566	**0.99574**	0.89001	0.92781	0.96851	**0.99574**
Castle	MS-SSIM	↑	0.98327	**0.98847**	0.98675	0.96927	0.97122	0.97759	**0.98705**
Alex	PSNR	↑	31.311	**43.374**	41.681	26.367	29.655	30.931	**44.867**
Dmitriy	PSNR	↑	37.805	**39.852**	39.678	24.194	27.319	29.623	**39.482**
Castle	PSNR	↑	27.811	**33.088**	32.814	27.544	27.805	29.346	**31.759**
Alex	VMAF	↑	94.689	**99.370**	99.164	74.482	92.388	93.585	**99.513**
Dmitriy	VMAF	↑	99.553	**99.807**	99.572	67.088	83.200	92.782	**99.785**
Castle	VMAF	↑	82.543	**88.677**	85.411	69.725	72.663	79.461	**87.314**
Alex	SAD	↓	5.251	**2.382**	2.571	10.841	6.125	9.620	**1.198**
Dmitriy	SAD	↓	6.966	1.496	**1.461**	14.789	9.298	10.992	**1.331**
Castle	SAD	↓	28.057	**3.916**	6.302	50.792	18.620	22.795	**10.780**
Alex	MSE (10^3)	↓	3.192	**1.151**	2.371	22.090	25.297	27.231	**0.452**
Dmitriy	MSE (10^3)	↓	9.129	0.898	**0.745**	48.158	43.058	33.835	**1.084**
Castle	MSE (10^3)	↓	4.310	**0.188**	0.406	31.412	11.084	7.410	**1.273**
Alex	Grad (10^{-1})	↓	93.36	**30.79**	39.29	683.53	877.07	765.106	**18.29**
Dmitriy	Grad (10^{-1})	↓	485.13	**45.68**	54.49	2422.75	2248.97	1740.41	**91.46**
Castle	Grad (10^{-1})	↓	381.44	**53.03**	79.32	7896.57	3594.16	2483.34	**242.89**
Alex	Conn (10^{-2})	↓	20.148	**9.369**	12.730	80.989	61.996	97.17	**4.132**
Dmitriy	Conn (10^{-2})	↓	51.299	8.431	**7.404**	128.131	94.101	110.85	**10.760**
Castle	Conn (10^{-2})	↓	157.124	**12.545**	26.446	459.208	184.359	211.11	**71.161**
Trolls	MAD	↓	2.5319	**1.0696**	1.2605	1.0284	1.1385	0.9393	**0.8866**

Table 2: Seven matting algorithms are evaluated on three ground truth datasets: Alex, Dmitriy, and Castle which are composited with real-world backgrounds as can be seen in the left columns of Figure 10 and on the Trolls sequence captured by our demonstrator (right columns in Figure 10). The best results of the two groups, using a trimap or using 2 frames as input, are marked in bold and the best results globally are underlined. There are three experiments: (1) evaluation is done on composition level with a checkerboard background. Here alpha and predictions are jointly measured using MS-SSIM, PSNR and VMAF scores. (2) alpha predictions are measured independently using SAD, MSE, gradient and connectivity. (3) there is no ground truth for the demonstrator sequence Trolls. However, the mean alpha deviation (MAD), as in Eq. 5, measures alpha value consistency across the predicted video sequence. The experiments show that FBA [10], SIM [41] and our algorithm perform similarly and achieve top scores. The two columns on the right show a short ablation study on the impact of our motion augmentation ours$_{ma}$ during training versus a static version ours$_{sa}$.

aberration as described in Figure 2 and 3 is not explicitly modeled in our approach, the joint, temporally mixed latent space in our one-encoder-dual-decoder in combination with the motion augmentation seems to increase temporal consistency. Although this is speculative, we believe that the effect of aberration and foreground motion are related in our feature space. Since neural networks interpolate and create local smoothness between known data, this could explain that robustness with respect to motion also has a positive effect on aberration compensation.

5 CONCLUSION

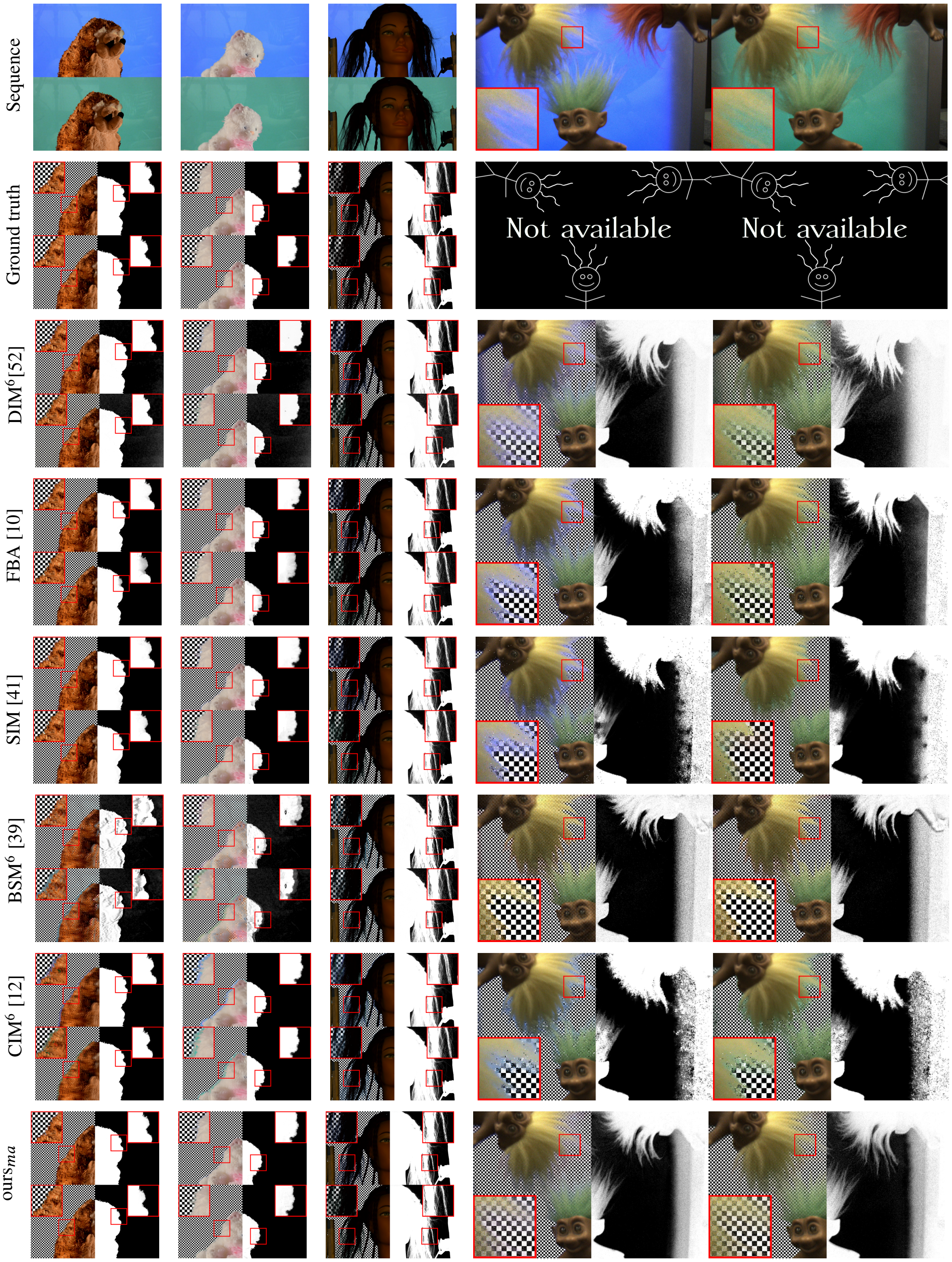
This paper presents a novel neural network-based dual-backdrop duplex matting system that creates high-quality alpha as well as foreground color predictions. It is temporally consistent, unaffected by noise, and shows superior color spill compensation. We compare our approach to a trimap-guided twin method that is trained and tested on the same datasets. In this experiment we clearly show that temporal backdrop duplex matting achieves superior results to the trimap-based approach. In addition, we propose a hardware set, that is actor friendly and can potentially be used in upcoming LED video wall production studios.

In the future, we intend to research the impact of chromatic aberration in more detail. This effect could be explicitly modeled in the dataset generation and augmentation, in order to further increase temporal consistency in alpha matting. Furthermore, we will investigate adaptation to view-dependent variations of the backings by camera motion prediction as in Dockhorn and Kruse [7].

ACKNOWLEDGMENTS

This work was supported by the Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor (grant no. 01DD20003) and the AI service center KISSKI (grant no. 01IS22093C), the Center for Digital Innovations (ZDIN) and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122).

[6] reimplementation



(a) Dmitriy (b) Alex (c) Castle (d) Trolls with blue background (e) Trolls with green background

Figure 10: Qualitative comparison of five matting methods. Please zoom in for better comparison.

REFERENCES

[1] 2022. Stagecraft by Industrial Light and Magic. https://www.ilm.com/stagecraft/. Accessed: 2022-01-07.

[2] Aseem Agarwala, Aaron Hertzmann, David Salesin, and Steven Seitz. 2004. Keyframe-based tracking of rotoscoping and animation. *ACM Trans. Graph.* 23 (08 2004), 584–591. https://doi.org/10.1145/1015706.1015764

[3] Glen Akins. 2014. RGB LED Panel Driver Tutorial. https://bikerglen.com/projects/lighting/led-panel-1up/

[4] Marcos H. Backes and Manuel M. Oliveira. 2019. A PatchMatch-based Approach for Matte Propagation in Videos. *Computer Graphics Forum* 38, 7 (2019), 651–662. https://doi.org/10.1111/cgf.13868 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13868

[5] Guangying Cao, Jianwei Li, Xiaowu Chen, and Zhiqiang He. 2019. Patch-Based Self-Adaptive Matting for High-Resolution Image and Video. *Vis. Comput.* 35, 1 (Jan. 2019), 133–147. https://doi.org/10.1007/s00371-017-1424-3

[6] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. 2001. A Bayesian Approach to Digital Matting. In *Proceedings of IEEE CVPR 2001* (Kauai, Hawaii), Vol. 2. IEEE Computer Society, 264–271.

[7] Alexander Dockhorn and Rudolf Kruse. 2020. Forward Model Learning for Motion Control Tasks. In *2020 IEEE 10th International Conference on Intelligent Systems (IS)*. 1–5. https://doi.org/10.1109/IS48319.2020.9199978

[8] Mikhail Erofeev, Yury Gitman, Dmitriy Vatolin, Alexey Fedorov, and Jue Wang. 2015. Perceptually Motivated Benchmark for Video Matting. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Article 99, 12 pages. https://doi.org/10.5244/C.29.99

[9] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. 2015. FlowNet: Learning Optical Flow with Convolutional Networks. arXiv:1504.06852 [cs.CV]

[10] Marco Forte and François Pitié. 2020. *F*, *B*, Alpha Matting. *CoRR* abs/2003.07711 (2020).

[11] Oliver Grau, Tim Pullen, and Graham A. Thomas. 2004. A combined studio production system for 3-D capturing of live action and immersive actor feedback. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 3 (2004), 370–380. https://doi.org/10.1109/TCSVT.2004.823397

[12] Anselm Grundhöfer, Daniel Kurz, Sebastian Thiele, and Oliver Bimber. 2010. Color Invariant Chroma Keying and Color Spill Neutralization For Dynamic Scenes and Cameras. *Vis. Comput.* 26, 9 (Sept. 2010), 1167–1176. https://doi.org/10.1007/s00371-010-0464-8

[13] Vikas Gupta and Shanmuganathan Raman. 2017. Automatic Trimap Generation for Image Matting. *CoRR* abs/1707.00333 (2017). arXiv:1707.00333 http://arxiv.org/abs/1707.00333;https://dblp.org/rec/journals/corr/GuptaR17.bib

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[15] Qiqi Hou and Feng Liu. 2019. Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation. *CoRR* abs/1909.09725 (2019). arXiv:1909.09725

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. http://lmb.informatik.uni-freiburg.de/Publications/2017/IMSKDB17

[17] Philip Torr James Thewlis, Shuai Zheng and Andrea Vedaldi. 2016. Fully-trainable deep matching. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Article 145, 12 pages. https://doi.org/10.5244/C.30.145

[18] Neel Joshi, Wojciech Matusik, and Shai Avidan. 2006. Natural Video Matting Using Camera Arrays. *ACM Trans. Graph.* 25, 3 (jul 2006), 779–786. https://doi.org/10.1145/1141911.1141955

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

[20] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W. H. Lau. 2022. MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition. *2022 Conference on Artificial Intelligence AAAI* (2022). https://ojs.aaai.org/index.php/AAAI/article/view/19999

[21] Jan Tore Korneliussen and Keigo Hirakawa. 2014. Camera Processing With Chromatic Aberration. *IEEE Transactions on Image Processing* 23 (2014), 4539–4552.

[22] Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. 2010. Temporally Coherent Video Matting. *Graph. Models* 72, 3 (May 2010), 25–33. https://doi.org/10.1016/j.gmod.2010.03.001

[23] Jizhizi Li, Sihan Ma, Jing Zhang, and Dacheng Tao. 2021. Privacy-Preserving Portrait Matting. arXiv:2104.14222 [cs.CV]

[24] Jizhizi Li, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. 2020. End-to-end Animal Image Matting. arXiv:2010.16188 [cs.CV]

[25] Yaoyi Li, Qingyao Xu, and Hongtao Lu. 2020. Hierarchical Opacity Propagation for Image Matting. arXiv:2004.03249 [cs.CV]

[26] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara. 2016. Toward a practical perceptual video quality metric. *The Netflix Tech Blog* 6, 2 (2016).

[27] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. 2022. Robust High-Resolution Video Matting with Temporal Guidance. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2022), 3132–3141.

[28] Morgan McGuire and Wojciech Matusik. 2006. Real-time triangulation matting using passive polarization. In *Presented at the SIGGRAPH 2006 Sketches program* (Boston, Massachusetts). ACM, New York, NY, USA, 88. https://doi.org/10.1145/1179849.1179959

[29] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John F. Hughes, and Frédo Durand. 2005. Defocus Video Matting. *ACM Trans. Graph.* 24, 3 (jul 2005), 567–576. https://doi.org/10.1145/1073204.1073231

[30] Juan-Manuel Pérez-Rúa, Ondrej Miksik, Philip. H. S. Torr, and Patrick Pérez. 2020. ROAM: A Rich Object Appearance Model with Application to Rotoscoping. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 8 (2020), 1996–2010. https://doi.org/10.1109/TPAMI.2019.2904963

[31] Yu Qiao, Yuhao Liu, Xin Yang, Dongsheng Zhou, Mingliang Xu, Qiang Zhang, and Xiaopeng Wei. 2020. Attention-Guided Hierarchical Structure Aggregation for Image Matting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. 2015. Deep Convolutional Matching. *CoRR* abs/1506.07656 (2015). arXiv:1506.07656 http://arxiv.org/abs/1506.07656

[33] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. 2009. A perceptually motivated online benchmark for image matting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1826–1833. https://doi.org/10.1109/CVPR.2009.5206503

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (LNCS, Vol. 9351)*. Springer, 234–241.

[35] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2020. Background Matting: The World Is Your Green Screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[36] Ehsan Shahrian, Brian Price, Scott Cohen, and Deepu Rajan. 2014. Temporally coherent and spatially accurate video matting. *Computer Graphics Forum* 33, 2 (2014), 381–390. https://doi.org/10.1111/cgf.12297 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12297

[37] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. 2016. Deep Automatic Portrait Matting. In *ECCV*.

[38] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[39] Alvy Ray Smith and James F. Blinn. 1996. Blue Screen Matting. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA, 259–268. https://doi.org/10.1145/237170.237263

[40] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. 2006. Flash Matting. *ACM Trans. Graph.* 25, 3 (jul 2006), 772–778. https://doi.org/10.1145/1141911.1141954

[41] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. 2021. Semantic Image Matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[42] Yanan Sun, Guanzhi Wang, Qiao Gu, Chi-Keung Tang, and Yu-Wing Tai. 2021. Deep Video Matting via Spatio-Temporal Alignment and Aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 6975–6984. https://doi.org/10.1109/CVPR46437.2021.00690

[43] Chin-Hung Teng, Yun-Hsuan Liao, Yi-Chia Chou, and Sih-Yu Lin. [n. d.]. Removing blue screen background under non-uniform illumination. In *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*. 307–308. https://doi.org/10.1109/ICCE-China.2017.7991118

[44] Suramya Tomar. 2006. Converting video formats with FFmpeg. *Linux Journal* 2006, 146 (2006), 10.

[45] Aviv Tzidon and Dekel Tzidon. U.S. Patent 5,886,747, Mar. 1999.

[46] Borja Vidal. 2012. Chroma Key Visual Feedback Based on Non-Retroreflective Polarized Reflection in Retroreflective Screens. *IEEE Transactions on Broadcasting* 58, 1 (2012), 144–150. https://doi.org/10.1109/TBC.2011.2174275

[47] Borja Vidal and Juan A. Lafuente. 2016. Chroma key without color restrictions based on asynchronous amplitude modulation of background illumination on retroreflective screens. *Journal of Electronic Imaging* 25, 2 (2016), 1 – 5. https://doi.org/10.1117/1.JEI.25.2.023009

[48] Petro Vlahos. U.S. Patent 4,007,487, Feb. 1977.

[49] Jue Wang and Michael F. Cohen. 2007. Image and Video Matting: A Survey. *Found. Trends. Comput. Graph. Vis.* 3, 2 (Jan. 2007), 97–175. https://doi.org/10.1561/0600000019

[50] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, Vol. 2. 1398–1402 Vol.2. https://doi.org/10.1109/ACSSC.2003.1292216

[51] Yuxin Wu and Kaiming He. 2020. Group Normalization. *Int. J. Comput. Vis.* 128, 3 (2020), 742–755. https://doi.org/10.1007/s11263-019-01198-w

[52] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. [n. d.]. Deep Image Matting. http://arxiv.org/abs/1703.03872;https://www.bibsonomy.org/bibtex/2feddb5247129d8477592afd493d64f8f/axel.vogler Computer Vision and Pattern Recognition (CVPR) 2017.

[53] Haichao Yu, Ning Xu, Zilong Huang, Yuqian Zhou, and Humphrey Shi. 2020. High-Resolution Deep Image Matting. arXiv:2009.06613 [cs.CV]

[54] Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-Sheng Hua 0001, Hujun Bao, Qixing Huang, and Weiwei Xu. 2021. Attention-guided Temporally Coherent Video Object Matting. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*. ACM, 5128–5137. https://doi.org/10.1145/3474085.3475623

[55] Yuanjie Zheng and Chandra Kambhamettu. 2009. Learning based digital matting. In *IEEE 12th International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 889–896. https://doi.org/10.1109/ICCV.2009.5459326

[56] Yuhongze Zhou, Liguang Zhou, Tin Lun Lam, and Yangsheng Xu. 2021. Human Perception Modeling for Automatic Natural Image Matting. *ArXiv* abs/2103.17020 (2021).