Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation

Qiqi Hou
Portland State University
qiqi2@pdx.edu

Feng Liu
Portland State University
fliu@cs.pdx.edu

Abstract

Natural image matting is an important problem in computer vision and graphics. It is an ill-posed problem when only an input image is available without any external information. While the recent deep learning approaches have shown promising results, they only estimate the alpha matte. This paper presents a context-aware natural image matting method for simultaneous foreground and alpha matte estimation. Our method employs two encoder networks to extract essential information for matting. Particularly, we use a matting encoder to learn local features and a context encoder to obtain more global context information. We concatenate the outputs from these two encoders and feed them into decoder networks to simultaneously estimate the foreground and alpha matte. To train this whole deep neural network, we employ both the standard Laplacian loss and the feature loss: the former helps to achieve high numerical performance while the latter leads to more perceptually plausible results. We also report several data augmentation strategies that greatly improve the network’s generalization performance. Our qualitative and quantitative experiments show that our method enables high-quality matting for a single natural image.

1. Introduction

Natural image matting is the problem of estimating the foreground image and the corresponding alpha matte from an input image. It is a critical step of image composition, which is widely used in image and video production. Without any external information, matting is a seriously ill-posed problem. In practice, most existing matting methods take a trimap as input; however, matting is still underconstrained in the undefined area in the trimap.

Traditional methods solve the matting problem by inferring the alpha matte information in the undefined area from those in the defined areas [51]. For instance, the matte values in the undefined areas can be computed by sampling the color or texture distribution of the known foreground and background areas and optimizing a carefully defined metric, such as the likelihood of the foreground, background, and alpha values [11, 18, 19, 49, 50]. While these methods provide promising results and some of them are incorporated into commercial tools, single natural image matting is still a challenging problem as these methods rely on the distinctive appearance of the foreground and background areas, such as their local or global color distribution.

Our research is inspired by the recent deep learning approaches to image matting. These deep matting approaches, such as [4, 33, 52] take an input image and the corresponding user-provided trimap as input and output an alpha map. They are shown robust for many challenging scenarios. These methods, however, only output the alpha map without the foreground or background image.

This paper presents a deep image matting method that simultaneously estimate the alpha map and the foreground image. Our method explores both local image and global context information for high-quality matting. This is inspired by the success of non-deep learning-based matting approaches that combines the global sampling and local propagation strategies [2, 6, 7, 20, 19]. Specifically, we designed a two-encoder-two-decoder fully convolutional neural network for context-aware simultaneous foreground image and alpha map estimation. The matting encoder learns to extract the local features while the context encoder learns more global features. We concatenate the features from these two encoders and feed them to an alpha decoder and a foreground decoder to estimate the alpha map and the corresponding foreground image simultaneously.

We explore a Laplacian loss and the feature loss to train our deep matting neural network. We found that the Laplacian loss enables our network to achieve the state-of-the-art numerical performance while the feature loss leads to more perceptually plausible matting results. We also found that some data augmentation methods are particularly helpful for our neural network to generalize to real-world images although our network is trained on a synthetic dataset pro-
To our best knowledge, this paper contributes the first deep matting method that enables simultaneous foreground and alpha estimation. Both our qualitative and quantitative experiments demonstrate that our method is able to generate state-of-the-art matting results on challenging real-world examples, as shown in Figure 1. We attribute the success of our method to 1) the integration of local visual features and global context information, 2) the combination of the Laplacian and feature loss, and 3) various effective data augmentation strategies that help generalizing our method to a wide variety of challenging real-world images.

2. Related Work

Image matting assumes that an image $I$ is a linear composition of a foreground image $F$ and a background image $B$ according to an alpha map $\alpha$ as follows:

$$I = \alpha F + (1 - \alpha)B$$

Given the input image $I$, image matting aims to recover $F$, $B$, and $\alpha$. Most of existing matting methods require a user-provided trimap that specifies known foreground and background areas, as well as an undefined area. In this way, matting is reduced to solving for the foreground, background, and alpha values in the undefined area. Given only the input image $I$ and the trimap, matting is a seriously ill-posed problem. A rich literature exists for matting. These methods infer the matte information for the undefined area from the known foreground and background according to the trimap. They either propagate the matte information from the neighboring foreground or background areas to the unknown areas [2] [6] [7] [20] [26] [27] [28] [46], or more globally sample the appearance information of the known foreground and background and use them to optimize for the matting in the unknown area [11] [18] [19] [49] [50]. There are also methods that combine the local propagation strategy and the global sampling strategy to achieve more reliable results [2] [6] [7] [20] [19]. Wang and Cohen provided a good survey on these traditional image matting algorithms [51]. Our design of a double-encoder-double-decoder network to learn to estimate local and global context information is inspired by these hybrid methods.

Our work is most relevant to the recent deep learning approaches to image matting. Shen et al. trained a dedicated deep convolutional neural network for portrait matting [43]. Their method first employs a deep neural network to generate the trimap of a portrait image and then feeds it to an off-the-shelf matting method, namely the Closed-form Matting algorithm [27], to obtain the final matting result. Cho et al. developed a deep matting method that takes the matting results from the Closed-form Matting algorithm [27] and the KNN Matting algorithm [6] as input, and refine it using a deep neural network [8] [9]. Xu et al. developed a large-scale synthetic image matting dataset and used it to train a two-stage deep neural network for alpha matting. Their method produced high-quality matting results for both synthetic and real-world images [52]. Lutz et al. explores generative adversarial networks to achieve high-quality natural image matting [53]. In their recent work, Chen et al. addressed a difficult case of image matting, transparent object matting. By considering transparent object matting as a refractive flow estimation problem, they developed a two-stage neural network to estimate the refractive flow from only one input image for transparent object matting [4]. While these methods are able to estimate high-quality alpha maps, they do not generate the foreground component. Our work builds upon these deep learning methods and simultaneously estimate the foreground image and the alpha map, thus providing a
complete solution to image matting. Our network learns to extract both local visual features and global context information to obtain high-quality image matting.

3. Context-Aware Image Matting

Our method takes an image $I$ and a user-specified trimap $T$ as input and aims to estimate the foreground $F$ and the corresponding alpha map $\alpha$, thus providing a full solution to matting. With the foreground and the alpha map, we can directly compute the background according to Equation 1.

We design a context-aware two-encoder-two-decoder deep neural network to simultaneously estimate the foreground and the alpha map, as shown in Figure 2. The outputs of the two encoders are concatenated and fed to the two decoder to generate the foreground and the alpha map, respectively. The two-encoder design of the network is inspired by the success of traditional matting algorithms that combine the local propagation and global sampling strategies for robust image matting [2, 6, 19, 20]. Specifically, the matting encoder is designed to learn to extract local features that are required to capture final image structures, such as hairs, while the context encoder learns to estimate more global context information that is helpful to disambiguate the foreground and background when they are similar to each other locally. Below we describe the encoders and decoders in more detail.

Matting encoder. We adopt the modified version of the Xception 65 architecture [10] from the deeplab v3+ [5] and set the down-sampling factor as 4 by setting the entropy flow's block2 and block3's stride as 1. This modification enables the middle flow to have a big spatial resolution. While traditional classification models [10, 21, 24, 41, 44] more aggressively compromise the spatial resolution to have a large valid receptive field, we use such a smaller down-sampling factor to retain sufficient spatial information that is important for the task of matting to capture fine image structures. Meanwhile, there is a trade-off between the computation/memory cost and spatial resolution. We empirically find that the down-sampling factor of 4 can get good matting results and cost a relatively small amount of computation and memory. We use skip connections to use features from the earlier layers as shown in Figure 2.

Context encoder. We also adopt the Xception 65 architecture [10] from [5]. Compared to the matting encoder, we use a much larger down-sampling factor of 16 to capture more global contextual information. We bilinearly upsample the final features by a factor of 4 so that the context features are of the same size as the local matting features from the matting encoder.

Alpha decoder and foreground decoder have the same network architecture. Specifically, we first bilinearly upsample the concatenated features from the encoders by a factor of 2 and then combine them with the intermediate features from the context encoder using a skip connection as shown in Figure 2. This is followed by two $3 \times 3$ convolutional layers with 64 channels. We repeat this process twice so that each decoder outputs the foreground image and the alpha map with the same size as the input image.

3.1. Loss functions

We compute the loss over both the alpha map and the foreground image. We explore a range of loss functions to
train our network. Below we describe them one by one.

We use a Laplacian loss \(L_{lap}^{\alpha} \) to measure the difference between the predicted alpha map \( \alpha \) and its ground truth \( \hat{\alpha} \).

\[
L_{lap}^{\alpha} = \sum_{i=1}^{5} 2^{i-1} ||L^i(\hat{\alpha}) - L^i(\alpha)||_1, \tag{2}
\]

where \(L^i(\alpha)\) indicates the \(i^{th}\) level of the Laplacian pyramid of the alpha map. This loss function measures the differences between the ground-truth pyramid representations and captures the local and global difference. We scale the contribution of a Laplacian level according to its spatial size.

We also use the feature loss to measure the perceptual quality of the alpha map. The feature loss, based on the differences between the high-level features extracted from a pre-trained convolutional neural network, has been shown effective in generating perceptually high-quality images in many image enhancement and synthesis tasks \([14, 25, 35, 36, 40, 53, 55]\). However, it is difficult to directly measure the perceptual quality of an alpha map. Our solution is to composite the ground-truth foreground image onto the black background using the alpha map and then measure the perceptual quality of the composition result as follows.

\[
L_{f} = \sum_{layer} \|\phi_{layer}(\hat{\alpha} * \hat{F}) - \phi_{layer}(\alpha * F)\|_2^2, \tag{3}
\]

where \(\hat{F}\) indicates the ground truth foreground and \(\phi_{layer}\) indicates the features output by the layer in a pre-trained VGG16 network \([44]\). Our method uses \([conv1_2, conv2_2, conv3_3, conv4_3]\) to compute the features.

We follow the same setting to calculate the feature loss for the predicted foreground image. Here the feature loss \(L_{f}^{\hat{F}}\) is computed on the composition result using the ground-truth alpha map with the foreground image as follows.

\[
L_{f}^{\hat{F}} = \sum_{layer} \|\phi_{layer}(\hat{\alpha} * \hat{F}) - \phi_{layer}(\alpha * F)\|_2^2, \tag{4}
\]

We also use the standard \(\ell_1\) loss for the predicted foreground \(F\). We only calculate the loss where the foreground is visible, in other words, the ground truth alpha matte is bigger than 0,

\[
L_{1} = ||1(\hat{\alpha} > 0) * (\hat{F} - F)||_1, \tag{5}
\]

where \(1\) is an indicator function that takes 1 if the statement is true and 0 otherwise.

Finally, we apply the standard \(\ell_2\) regularization loss to all the convolutional layers. We will examine these loss functions in our experiments (Section 4).

3.2. Training

We initialize our neural network with pre-trained models from \([5]\). We use TensorFlow to train our neural network. Similar to \([5]\), we use the “poly” learning rate policy to train our network, where \(lr = lr_{init}(1 - \frac{iter}{max_iter})^{power}\) with \(lr_{init} = 7 \times 10^{-4}\) and \(power = 0.9\). We use a minibatch size of 6 and train the neural network for 1 million iterations for models (1-3) in Table 1. We fine-tune models (4-9) based on the pretrained model (3) with \(10^5\) iterations with \(lr_{init} = 10^{-4}\).

Training dataset. We train our network using the matting dataset shared by Xu et al. \([52]\). This dataset contains 431 training images with the corresponding alpha maps and the foreground images. We create the training samples in a similar way to Xu et al. Specifically, we composite the foreground image onto a randomly selected background image from MS-COCO dataset \([50]\). We down-sample the foreground image gradually by a factor of 0.9 until the short side is 600 pixels. If the source image’s short side is less than 600 pixels, we first scale it up to 780. In total, we generate 1957 scaled foreground image. Then we select image patches that contain unknown regions in the trimap. Specifically, we slide windows of size 600 × 600 on the full image with a stride of 5 pixels to get a large amount of candidate windows and remove patches where less than 10% pixels are unknown. Furthermore, since many patches overlap with each other significantly, we employ non-maximum suppression(NMS) to remove overlapping patches. Specifically, we set the NMS threshold as 0.3 and only keep the top 30 image patches with the highest unknown pixel percentages in each image. Figure 3 shows an example of selected image patches. In total, we obtain 9,507 600 × 600 foreground image patches. Finally, we create training samples of size 225 × 225 by randomly cropping the composited image with the following data augmentation operators.

Data augmentation. Following Xu et al. \([52]\), our training samples are obtained by compositing a foreground image and a background image using an alpha map. As reported in many papers \([1, 13, 17, 22, 23, 31, 52, 57, 47]\), many sub-
Figure 4. Data augmentation. In the composited image without any data augmentation (a), the foreground image contains some JPEG artifacts while the background is smooth, which produces a bias that will compromise the training of the network. Re-JPEGing introduces the artifacts to the foreground and the background to reduce the bias while Gaussian Blur does so by smoothing the high-frequency artifacts.

dete the artifacts, such as misaligned JPEG blocks, compression quantization artifacts, and resampling artifacts, can sometimes affect their methods a lot despite that the images look plausible to the human eyes. Some splice detection methods [11, 22, 23, 32, 37, 47] even build their algorithms based on such an observation. Directly training the network on the composited images without special augmentation may suffer from a similar problem and thus compromises the generalization capability of the trained network.

Therefore, besides the resizing augmentation used in Xu et al. [52], we follow the post-processing steps in the image splice detection methods [12, 23, 34] and use re-JPEGing and Gaussian blur to augment our training samples. These operators introduce subtle artifacts that are not visually noticeable but can make the network less bias to the small difference between the foreground and the background. As shown in Figure 4, the original background is smoother than the original foreground image. Therefore, it is possible that the network relies on this bias to differentiate the foreground from the background. Re-JPEGing and Gaussian blur can relieve this problem by introducing artifacts or remove these artifacts. For re-JPEGing, we keep 70% quality of the composited images. For Gaussian blur, we on-the-fly generate a Gaussian kernel with standard deviation in the range of [0, 3] and the kernel size in the range of [3, 5], and apply it to the composited image. We also randomly resize the composited image with a rate of between 0.5 and 1.

Besides, we also use some standard data augmentation operators. Specifically, we employ the gamma transforms to increase the color diversity. The gamma value is randomly selected from [0.2, 2]. We randomly flip the images horizontally. The trimap for each image is automatically generated by randomly dilating its corresponding ground truth alpha map in the range of [4, 25].

4. Experiments

We experiment with our methods on the synthetic Composition-1K dataset and a real-world matting image dataset, both of which are provided by Xu et al. [52]. As discussed in Section 3.2, our neural networks are all trained on the synthetic Composition-1K training set. We evaluate our models and compare to the state of the art methods on the Composition-1K testing set and the real-world matting image set. Specifically, the Composition-1K testing dataset contains 1000 composited images. They were generated by compositing 50 unique foreground images onto each of the 20 images from the PASCAL VOC 2012 dataset [15]. We used the code provided by Xu et al. [52] to generate these testing images. The real world image dataset contains 31 real world images pulled from the internet [52]. We conduct our user study on the real world images.

Since not all the methods produce both the foreground images and the alpha maps as the final matting results, we compare our methods to the state of the art on the alpha maps and the foreground images separately. Besides, we also report our user study and our ablation studies to more thoroughly evaluate our methods.

4.1. Evaluation on alpha maps

We compare our methods to both the state of the art non-deep learning methods, including Shared Matting [16], Learning Based Matting [54], Comprehensive Sampling [42], Global Matting [19], Closed-form Matting [27], KNN Matting [6], Three-layer Graph [29], Information-flow Matting [2], and recent deep learning matting approaches, including DCNN Matting [8], Deep Matting [52] and AlphaGan [33]. Table 1 reports the results on these methods as well as ours on the Composition-1K dataset. The results of the comparing methods are obtained

<table>
<thead>
<tr>
<th>Methods</th>
<th>SAD</th>
<th>MSE(10^5)</th>
<th>Grad</th>
<th>Conn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Matting [16]</td>
<td>128.9</td>
<td>91</td>
<td>120.5</td>
<td>135.3</td>
</tr>
<tr>
<td>Learning Based Matting [54]</td>
<td>113.9</td>
<td>48</td>
<td>91.6</td>
<td>122.2</td>
</tr>
<tr>
<td>Comprehensive Sampling [12]</td>
<td>143.8</td>
<td>71</td>
<td>102.2</td>
<td>142.7</td>
</tr>
<tr>
<td>Global Matting [19]</td>
<td>133.6</td>
<td>68</td>
<td>97.6</td>
<td>133.3</td>
</tr>
<tr>
<td>Closed-Form Matting [27]</td>
<td>168.1</td>
<td>91</td>
<td>126.9</td>
<td>167.9</td>
</tr>
<tr>
<td>KNN Matting [6]</td>
<td>175.4</td>
<td>103</td>
<td>124.1</td>
<td>176.4</td>
</tr>
<tr>
<td>DCNN Matting [8]</td>
<td>161.4</td>
<td>87</td>
<td>115.1</td>
<td>161.9</td>
</tr>
<tr>
<td>Three-layer Graph [29]</td>
<td>106.4</td>
<td>66</td>
<td>70</td>
<td>0.7</td>
</tr>
<tr>
<td>Deep Matting [52]</td>
<td>50.4</td>
<td>14</td>
<td>31.0</td>
<td>50.8</td>
</tr>
<tr>
<td>Information-flow Matting [2]</td>
<td>75.4</td>
<td>66</td>
<td>63.0</td>
<td>50.8</td>
</tr>
<tr>
<td>AlphaGan-Best [33]</td>
<td>52.4</td>
<td>30</td>
<td>38.0</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1. Alpha map results on the Composition-1K testing set.
As shown in Table 1, our two-encoder-two-decoder models (model 3-9) generate matting results with significantly smaller errors than the state of the art methods. To understand what contributes to this improvement, we evaluated on a baseline method (model 2) that removes the context encoder and found that this baseline model performs much worse according to all the four metrics. Therefore, the improvement can be mainly attributed to the use of our two encoders to capture both local visual features for fine details and more global contextual information to disambiguate the locally similar foreground and background.

As shown in Table 2, our method reduces the error by a large margin. As existing deep learning methods only output alpha maps, we compare to three representative non-deep learning matting methods, namely Global Matting [19], Closed-Form Matting [27] and KNN Matting [6], on how well foreground images can be extracted from single input images on the Composition-1K dataset. We calculate the SAD and MSE of $\alpha \odot F$ following the previous work [38]. As shown in Table 2, our method reduces the error by a large margin.

### 4.3. Ablation study

As discussed in Section 4.1, our two-encoder structure brings in the major performance improvement. Besides, we found that proper loss functions and data augmentations are also important to obtain high-quality matting results and help generalizing to real-world images.

**Loss functions.** As shown in Table 1, our model (2) with the Laplacian loss $L_{lap}^{\alpha}$ generates more numerically accurate results than our model (1) with the loss used in Deep Matting [52]. Our model (3) generates better result compared to the model (4) with both the Laplacian loss and the feature loss $L_{F}^{\alpha}$. On the other hand, the feature loss enables our model (4) to generate visually better results that keep more final structures than our model (3), as shown in the last example in Figure 6. This is consistent with many other works on image synthesis tasks that the feature loss tends to produce perceptually better results (often at the expense of the numerical performance) [3, 14, 25, 35, 40, 53, 55].

When training our network with both the foreground decoder and the alpha decoder, color loss functions, namely $L_{c}^{\alpha}$ and $L_{F}^{\alpha}$, are naturally needed. By comparing models (4) and (6), (5) and (7) in Table 1, we find that these color losses can improve the alpha map estimation slightly. This is in part because the color and the alpha decoders share

### Table 2. The foreground result on the Composition-1K dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SAD</th>
<th>MSE($10^{15}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Matting [19]</td>
<td>220.39</td>
<td>36.29</td>
</tr>
<tr>
<td>Closed-Form Matting [27]</td>
<td>254.15</td>
<td>40.89</td>
</tr>
<tr>
<td>KNN Matting [6]</td>
<td>281.92</td>
<td>36.29</td>
</tr>
<tr>
<td>(6) ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$ + $L_{1}^{\alpha}$ + $L_{F}^{\alpha}$ + DA</td>
<td>61.72</td>
<td>3.24</td>
</tr>
<tr>
<td>(7) ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$ + $L_{1}^{\alpha}$ + $L_{F}^{\alpha}$ + DA</td>
<td>94.41</td>
<td>8.67</td>
</tr>
<tr>
<td>(8) ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$ + $L_{1}^{\alpha}$ + $L_{F}^{\alpha}$ + DA - ReJPEGing</td>
<td>73.79</td>
<td>4.96</td>
</tr>
<tr>
<td>(9) ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$ + $L_{1}^{\alpha}$ + $L_{F}^{\alpha}$ + DA - GaussianBlur</td>
<td>85.8</td>
<td>7.10</td>
</tr>
</tbody>
</table>

### Table 3. Parameter numbers of our models and their performance on the Composition-1K dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th># of Parameters</th>
<th>SAD</th>
<th>MSE</th>
<th>Grad</th>
<th>Conn</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME (model 2)</td>
<td>54.0 M</td>
<td>43.9</td>
<td>11.8</td>
<td>20.6</td>
<td>41.6</td>
</tr>
<tr>
<td>ME (deeper model 2)</td>
<td>117.0 M</td>
<td>43.7</td>
<td>11.0</td>
<td>21.2</td>
<td>42.6</td>
</tr>
<tr>
<td>ME + CE (model 3)</td>
<td>107.5 M</td>
<td>35.8</td>
<td>8.2</td>
<td>17.3</td>
<td>33.2</td>
</tr>
</tbody>
</table>

### Table 4. Comparison of visual quality on the real-world dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean score</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME + CE + $L_{lap}^{\alpha}$</td>
<td>4.64</td>
<td>0.42</td>
</tr>
<tr>
<td>ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$</td>
<td>4.69</td>
<td>0.40</td>
</tr>
<tr>
<td>ME + CE + $L_{lap}^{\alpha}$ + $L_{F}^{\alpha}$ + DA</td>
<td>5.03</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Deep Matting work [52], we examine our method’s sensitivity to the trimap size by dilating the ground-truth to a range of sizes. As illustrated in Figure 5 our method is stable to the trimap sizes. Note, the scores of comparing methods were obtained from [52].

### 4.2. Evaluation on foreground images

As existing deep learning methods only output alpha maps, we compare to three representative non-deep learning matting methods, namely Global Matting [19], Closed-Form Matting [27] and KNN Matting [6], on how well foreground images can be extracted from single input images on the Composition-1K dataset. We calculate the SAD and MSE of $\alpha \odot F$ following the previous work [38]. As shown in Table 2, our method reduces the error by a large margin.
the same learned features, and the tasks of foreground color estimation and alpha map estimation are relevant.

**Data augmentation.** As shown in Table 1 data augmentations, such as ReJPEGing and Gaussian blur, can greatly increase the errors of our methods on the Composition-1k testing dataset. On the other hand, we found that these data augmentations can greatly improve the generalization of our trained networks on the real world images. As shown in Figure 6, when trained with these data augmentation strategies, our models can maintain more fine details, such as hairs. Since these real world examples do not have ground truth, to obtain objective scores of these results, we evaluate the quality of composition results using our matting results. Specifically, we composite the foreground objects in source images onto some external background images using our matting results and then measure the visual quality of the composition results using the NIMA quality assessment algorithm [48]. As reported in Table 4, our data augmentation algorithms are helpful. We also test our methods on the Spectral Matting dataset [28] with the known ground truth. This dataset is generated by photographing dolls in front of a computer monitor displaying seven different background images. The trimap is generated by dilating the alpha map by 20 pixels by alpha map denoising. Our method with DA outperforms our method without DA significantly according to most of the metrics: 3.58 vs 4.28 (SAD), 6.64 vs 9.05 (MSE), and 2.57 vs 3.19 (Conn), and slightly reduces the performance according to Grad: 2.04 vs 1.92.

### 4.4. User study

To further evaluate the quality of our results, we conducted a user study. We compared our method (model (7)) with three representative methods, including Deep Matting [52] and two state-of-the-art non-deep learning methods, Close-form Matting [27] and Global matting [19].

Our study used all the 31 real-world images from Xu et al. [52]. We used a similar protocol to Xu et al. [52] to produce the results for the study. For the methods except Deep Matting, we composite the predicted foreground and alpha map onto a blank background image. We use the black background or the white background randomly with the exceptions that for certain foreground images, a particular background color is not appropriate. For example, it is meaningless to composite the black hair onto a black background image, so for such an example, we choose to use the white background. Since Deep Matting does not output the foreground image, we composite the input image using the estimated alpha map as suggested in their paper. Therefore, the comparison between our results with those from Deep Matting should be interpreted with a grain of salt.

Our user study recruited 42 students with different backgrounds. None of them have previous experience with the matting task. Therefore, we conducted a training session for each participant before the formal study. Specifically, each of them was shown two real-world images. For each image, we showed two matting results from different meth-
ods without revealing which methods were used to generate these results. We then explained the differences between two results to the participant. This training session is helpful as the subtle difference in matting results was often difficult to spot for people with no prior matting experience.

In our study, we divided the 42 participants into three groups. Each group evaluated how our results compared to one of the three existing methods. In each trial, a participant was presented with a screen that only shows a source image and two corresponding matting results at a time. The participant could select which image to view by clicking the corresponding button or using the left or right key on the keyboard. In this way, the participant can flip between different images to examine the quality or compare the difference. In each trial, the participant was asked to choose a more accurate and realistic result between the two results. Each participant conducted 31 trials so that the results for all the 31 testing images are evaluated.

We calculated the percentage of the times that our results were preferred by the participants and then calculated the average and the standard deviation for each group. As reported in Table 5, more of our results are preferred by the participants than all the comparing methods. Figure 7 shows some examples in our study. They show that our method can better capture very fine structures like the hair in the first example even when the hair shares a similar color to the background. In the last example, our result not only keeps the delicate edge of the lace, which is lost in the other results, but also is free from the color bleeding problem where the blue background color contaminated the result.

5. Conclusion

This paper presented a context-aware deep matting method for simultaneously estimating the foreground and the alpha map from a single natural image. We developed a two-encoder-two-decoder neural network for this task. The two encoders were designed to capture both the local fine structures and the more global context information to disambiguate the foreground and background with a similar appearance. The two decoders output the foreground and the alpha map respectively. Our experiments showed that using the feature loss helps to obtain visually more pleasant matting results while the Laplacian loss tends to optimize the numerical performance. Our experiments also showed that dedicated data augmentation methods, such as Re-JPEGING and Gaussian blurring, are helpful to generalize the neural network trained on a synthetic dataset to handle real-world challenging matting tasks.

Acknowledgments. The source images in Figure 1 are used under a Creative Commons license from Flickr users Robbie Sproule, MEGA PISTOLO and Jeff Latimer. The background images in Figure 1 are from the MS-COCO dataset [30]. Source images used in Figure 2, 3, 4, 6, and 7 are from the matting dataset shared by Xu et al. [52]. We thank Nvidia for their GPU donation and Google for their cloud credits.

References


